**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

# BÁO CÁO THỰC NGHIỆM
# HỌC PHẦN: PHÂN TÍCH DỮ LIỆU LỚN

## ĐỀ TÀI: PHÂN TÍCH MÔ TẢ THỊ TRƯỜNG VÀNG VÀ DỰ BÁO GIÁ VÀNG BẰNG MÔ HÌNH HỒI QUY


**Giảng viên hướng dẫn:** TS. Nguyễn Mạnh Cường

**Lớp** : 20241IT6077003

**Nhóm thực hiện** : Nhóm 16

Tống Đăng Quang - 2022603783

Đỗ Trọng Hoàng - 2022604369

Nguyễn Huy Nhật - 2022605668


**Hà Nội – Năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

_____

# BÁO CÁO THỰC NGHIỆM
# HỌC PHẦN: PHÂN TÍCH DỮ LIỆU LỚN

## ĐỀ TÀI: PHÂN TÍCH MÔ TẢ THỊ TRƯỜNG VÀNG VÀ DỰ BÁO GIÁ VÀNG BẰNG MÔ HÌNH HỒI QUY


**Giảng viên hướng dẫn:** TS. Nguyễn Mạnh Cường

**Lớp** : 20241IT6077003

**Nhóm thực hiện** : Nhóm 16

Tống Đăng Quang - 2022603783

Đỗ Trọng Hoàng - 2022604369

Nguyễn Huy Nhật - 2022605668


**Hà Nội – Năm 2024**

# LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn quý thầy, cô trường Đại Học Công Nghiệp Hà Nội đã tận tình dạy dỗ chúng em, trong đó phải kể đến quý thầy cô trong Khoa Công nghệ thông tin đã tạo điều kiện để chúng em thực hiện đề tài tiểu luận.

Đặc biệt, chúng em xin chân thành cảm ơn giảng viên hướng dẫn – TS. Nguyễn Mạnh Cường đã tận tình giúp đỡ, hỗ trợ chúng em trong quá trình thực hiện đề tài. Cung cấp cho chúng em những kiến thức quý báu cũng như những lời khuyên hữu ích. Tạo động lực cho chúng em hoàn thành tốt nhiệm vụ của mình. Bên cạnh đó, chúng em cũng xin cảm ơn các bạn học viên trong Khoa Công nghệ thông tin đã đóng góp ý kiến giúp chúng em thực hiện đề tài đạt hiệu quả hơn.

Bài tiểu luận này đã giúp chúng em rèn luyện kỹ năng tư duy phân tích, xử lý dữ liệu và trình bày thông tin một cách có logic và rõ ràng. Chúng em hi vọng rằng những kiến thức và kinh nghiệm thu thập từ đề tài này sẽ tiếp tục hỗ trợ chúng em trong tương lai, không chỉ trong học tập mà còn trong sự nghiệp và cuộc sống.

Nhóm chúng em xin trân trọng cảm ơn!

**Nhóm học viên thực hiện**

Tống Đăng Quang

Đỗ Trọng Hoàng

Nguyễn Huy Nhật

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# LỜI NÓI ĐẦU

Thị trường vàng từ lâu đã được xem là một trong những thị trường tài chính quan trọng nhất, đóng vai trò là một hàn thử biểu của nền kinh tế toàn cầu. Với sự bất ổn của các biến động chính trị, kinh tế và các rủi ro tài chính, vàng thường trở thành nơi trú ẩn an toàn cho các nhà đầu tư. Chính vì vậy, việc hiểu rõ và dự báo xu hướng giá vàng không chỉ là một thách thức lý thú mà còn có ý nghĩa thực tiễn to lớn trong việc đưa ra các quyết định đầu tư chiến lược. Nghiên cứu về thị trường vàng giúp chúng ta nhìn nhận sâu sắc hơn về tác động của các yếu tố kinh tế vĩ mô, tâm lý nhà đầu tư, cũng như những sự kiện toàn cầu đến giá trị của tài sản này.

Đề tài "Phân tích mô tả thị trường vàng và dự báo giá vàng bằng mô hình hồi quy" được thực hiện với mục tiêu cung cấp cái nhìn toàn diện về thị trường vàng và khám phá các yếu tố chi phối sự biến động của giá vàng. Chúng em đã lựa chọn phương pháp hồi quy, một công cụ mạnh mẽ trong phân tích thống kê, để dự báo sự thay đổi của giá vàng dựa trên những dữ liệu quá khứ. Với đề tài này, chúng em không chỉ muốn trình bày bức tranh toàn cảnh về thị trường vàng mà còn muốn khám phá tiềm năng của các mô hình hồi quy trong việc dự đoán giá trị của tài sản tài chính.

Báo cáo này được cấu trúc thành bốn chương như sau:

*Chương 1: Tổng quan về đề tài*

Chương này giới thiệu bối cảnh nghiên cứu và mục tiêu của đề tài, nhấn mạnh tầm quan trọng của việc phân tích và dự báo giá vàng. Tình hình nghiên cứu trong và ngoài nước được trình bày, làm nền tảng cho việc áp dụng các phương pháp kỹ thuật trong các chương sau.

*Chương 2: Một số phương pháp*

Trình bày các phương pháp phân tích dữ liệu (mô tả và hồi quy) cùng công cụ chính là mô hình LSTM, Python và các thư viện hỗ trợ cũng được lựa chọn để thực hiện phân tích và dự báo.

*Chương 3: Thực nghiệm*

Chương này mô tả quy trình xử lý dữ liệu, phân tích mô tả, và triển khai mô hình LSTM để dự đoán giá vàng. Kết quả cho thấy mô hình hoạt động tốt,

với các chỉ số đánh giá như MAE và RMSE được sử dụng để kiểm tra độ chính xác.

*Chương 4: Xây dựng sản phẩm*

Trình bày về chương trình nhóm đã phát triển được, giúp người dùng nhập dữ liệu và nhận dự báo giá vàng. Giao diện sản phẩm được minh họa, cùng với đánh giá tính hiệu quả và các đề xuất cải tiến.

Thông qua việc thực hiện đề tài này, chúng em đã tích lũy được nhiều kiến thức bổ ích về lĩnh vực tài chính, đặc biệt là về thị trường vàng và cách mà các yếu tố kinh tế tác động lên giá trị của nó. Chúng em cũng đã có cơ hội ứng dụng các phương pháp phân tích dữ liệu và các mô hình dự báo hồi quy vào thực tiễn, từ đó nâng cao kỹ năng nghiên cứu và xử lý dữ liệu. Những kiến thức và kỹ năng này không chỉ hữu ích trong quá trình học tập mà còn có giá trị lâu dài đối với những ai mong muốn nghiên cứu sâu hơn về thị trường tài chính và đầu tư.

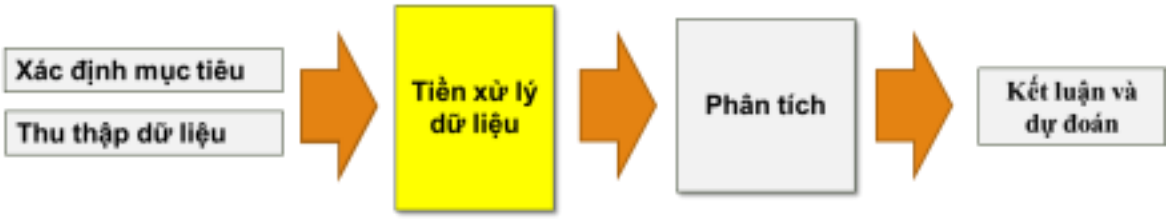
# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Tổng quan về phân tích dữ liệu

### *1.1.1. Phân tích dữ liệu là gì*

Phân tích dữ liệu là quá trình kiểm tra, làm sạch, chuyển đổi và mô hình hóa dữ liệu với mục tiêu khám phá thông tin hữu ích, đưa ra kết luận và hỗ trợ việc ra quyết định.

### *1.1.2. Quy trình phân tích dữ liệu*




*Hình 1.1. Quy trình phân tích dữ liệu [1]*

Quy trình phân tích dữ liệu thường bao gồm các bước chính:

- ***Xác định mục tiêu và thu thập dữ liệu:***
  + *Xác định mục tiêu*: là những kết quả cụ thể mà ta muốn đạt được thông qua việc xử lý và phân tích dữ liệu. Mục tiêu này xác định hướng đi và phạm vi của quá trình phân tích, giúp ta tập trung vào việc thu thập thông tin quan trọng và thực hiện các phân để đáp ứng các yêu cầu hoặc nhu cầu cụ thể.
  + *Thu thập dữ liệu*: là thu thập dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin, trang web, thiết bị cảm biến, và nhiều nguồn khác. Dữ liệu có thể là số liệu, văn bản, hình ảnh, hoặc âm thanh.
- ***Tiền xử lý dữ liệu:*** Dữ liệu thường không hoàn hảo và có thể chứa nhiễu, dữ liệu bị thiếu, hoặc không chính xác. Tiền xử lý dữ liệu bao gồm việc tóm lược dữ liệu, làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu, rút gọn dữ liệu và rời rạc hóa dữ liệu để chuẩn bị cho bước phân tích.
- ***Phân tích dữ liệu:*** Bước quan trọng này dựa vào kiến thức và kỹ thuật phân tích để tìm ra mối liên hệ và thông tin ẩn sau dữ liệu. Phân tích dữ liệu có thể sử dụng các phương pháp phân tích mô tả, phân tích hồi

quy, phân tích sự khác biệt, thống kê, machine learning, data mining, và nhiều kỹ thuật khác.

- ***Kết luận và dự đoán:*** Dựa trên phân tích và thông tin từ dữ liệu, chúng ta có thể rút ra kết luận, hiểu rõ hơn về tình hình, và thậm chí đưa ra dự đoán cho tương lai.

## 1.2. Tổng quan về bài toán phân tích mô tả

Phân tích mô tả là một phương pháp trong lĩnh vực thống kê và phân tích dữ liệu, nhằm mô tả và tóm tắt các đặc điểm chính của một tập dữ liệu một cách dễ hiểu và ngắn gọn. Mục tiêu của phân tích mô tả là giúp hiểu sâu hơn về dữ liệu mà chúng ta đang làm việc, nhận ra các đặc trưng quan trọng, và cung cấp một cái nhìn tổng quan về phân phối và biến đổi của dữ liệu. Phân tích mô tả thường bao gồm các khía cạnh sau:

+ *Thống kê tóm tắt:* Đây là các số liệu thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn, và phân vị. Các số liệu này giúp ta hiểu về trung tâm và phân tán của dữ liệu.
+ *Biểu đồ:* Biểu đồ thường được sử dụng để biểu diễn dữ liệu một cách trực quan. Các biểu đồ như biểu đồ cột, biểu đồ đường, biểu đồ hình tròn, và biểu đồ hộp giúp ta thấy được sự phân bố và xu hướng của dữ liệu.
+ *Phân phối dữ liệu:* Phân tích phân phối dữ liệu giúp ta hiểu về tỷ lệ xuất hiện của các giá trị khác nhau trong tập dữ liệu. Điều này có thể làm bằng cách tạo biểu đồ phân phối tần số hoặc xây dựng biểu đồ kernel density.
+ *Kiểm tra sự tương quan:* Phân tích mô tả cũng có thể liên quan đến việc kiểm tra sự tương quan giữa các biến. Điều này có thể thực hiện bằng cách sử dụng biểu đồ tương quan hoặc tính toán hệ số tương quan Pearson.
+ *Xác định điểm ngoại lệ:* Phân tích mô tả cũng giúp xác định các điểm dữ liệu ngoại lệ, tức là những giá trị rất khác biệt so với phần còn lại của dữ liệu.

+ *Tổng kết và nhận xét:* Cuối cùng, phân tích mô tả thường đi kèm với việc tổng kết và nhận xét về các đặc điểm quan trọng của dữ liệu, những mẫu thú vị, và những điểm mạnh và điểm yếu của tập dữ liệu.

Phân tích mô tả giúp xây dựng một cái nhìn sâu hơn về tập dữ liệu ban đầu và tạo nền tảng cho các phân tích tiếp theo như dự báo, phân tích hồi quy, hay machine learning.

## 1.3. Tổng quan về bài toán dự báo

### *1.3.1. Lịch sử về bài toán dự báo*

Bài toán dự báo có một lịch sử lâu đời và đã phát triển qua nhiều giai đoạn. Dưới đây là một cái nhìn tổng quan về lịch sử hình thành của bài toán dự báo:

*Thời kỳ tiền Công nghiệp (Trước thế kỷ 18):* Trong giai đoạn này, con người thường dự báo dựa trên kinh nghiệm và tri thức truyền đạt qua thế hệ. Dự báo chủ yếu dựa trên sự quan sát của thiên văn học, thời tiết, và các hiện tượng tự nhiên.

*Cách mạng Công nghiệp và thống kê (Thế kỷ 18 - 19):* Trong thời kỳ này, việc sử dụng số liệu và thống kê để dự báo đã trở nên phổ biến hơn. Những ý tưởng về xác suất và phân phối bắt đầu được áp dụng vào việc dự báo.

*Thế kỷ 20 và Kỹ thuật số hoá:* Sự phát triển của máy tính và kỹ thuật số hoá đã mở ra những cơ hội mới trong việc dự báo. Các phương pháp thống kê, mô hình hóa toán học, và kỹ thuật machine learning bắt đầu được sử dụng rộng rãi để dự báo trong nhiều lĩnh vực.

*Thống kê Bayes và Kỹ thuật Machine learning (Thế kỷ 20 - 21):* Thống kê Bayes và các kỹ thuật machine learning như học máy, học sâu, và học tăng cường đã thúc đẩy khả năng dự báo thông qua việc xử lý dữ liệu phức tạp và tìm ra các mẫu ẩn.

*Dự báo trong thời đại số hóa (Hiện nay):* Với sự gia tăng mạnh mẽ về khả năng tính toán, khối lượng dữ liệu khổng lồ, và sự phát triển của trí tuệ nhân tạo, bài toán dự báo đang trở nên càng quan trọng và phức tạp hơn. Các công nghệ mới như big data analytics, deep learning, và dự báo dựa trên mạng

xã hội đang mở ra nhiều cơ hội và thách thức mới trong lĩnh vực này. Trong suốt quá trình phát triển, bài toán dự báo đã chuyển từ việc dự đoán dựa trên sự quan sát đơn thuần đến việc sử dụng các phương pháp phức tạp để xác định mối quan hệ phức hợp và xu hướng từ dữ liệu. Lịch sử hình thành này thể hiện sự tiến bộ và tầm quan trọng của bài toán dự báo trong việc hỗ trợ quyết định và phát triển trong nhiều lĩnh vực.

Bài toán dự báo là một trong những thách thức quan trọng trong lĩnh vực phân tích dữ liệu, nơi chúng ta cố gắng dự đoán giá trị của một biến mục tiêu trong tương lai dựa trên dữ liệu lịch sử và các yếu tố ảnh hưởng. Mục tiêu chính của bài toán dự báo là xây dựng một mô hình có khả năng hiểu và ứng dụng các mẫu, xu hướng và quy luật từ dữ liệu để thực hiện việc dự đoán một cách chính xác và đáng tin cậy.

### *1.3.2. Tình hình nghiên cứu trong nước*

Bài toán dự báo có sự ảnh hưởng to lớn tại Việt Nam. Dự báo giúp cải thiện quản lý, định hình chiến lược, và tối ưu hóa tài nguyên trong nhiều lĩnh vực. Có một số điểm đáng chú ý về tình hình phân tích dữ liệu tại Việt Nam:

- *Phát triển đang ở giai đoạn đầu:* Trong một số lĩnh vực, bài toán dự báo tại Việt Nam đang ở giai đoạn đầu của sự phát triển. Việc áp dụng các phương pháp phân tích dữ liệu và dự báo mới còn đang được tìm hiểu và thí nghiệm.
- *Ứng dụng trong nông nghiệp và kinh tế:* Tại Việt Nam, dự báo có ứng dụng quan trọng trong nông nghiệp, nhằm dự đoán thời tiết, mùa màng, và nhu cầu năng lượng. Nó cũng được áp dụng trong kinh tế, dự báo tăng trưởng GDP, lạm phát và tỷ giá.
- *Thách thức từ dữ liệu:* Một thách thức cho việc dự báo tại Việt Nam là khả năng thu thập và quản lý dữ liệu chất lượng. Dữ liệu thường không đầy đủ và có thể gặp vấn đề về tính nhất quán và độ tin cậy.

### *1.3.3. Tình hình nghiên cứu ở nước ngoài*

Trong lĩnh vực nghiên cứu bài toán dự báo đã có một số công trình nghiên cứu ngoài nước có liên quan đến đề tài tiểu luận, ví dụ như: “Solar Forecast Reconciliation and Effects of Improved Base Forecasts” được đăng trên IEEE Xplore, tác giả: Gokhan Mert Yagli, Dazhi Yang, Dipti Srinivasan,

Monika. Đề tài nghiên cứu này trình bày về dự báo sản lượng điện mặt trời đóng vai trò quan trọng trong vận hành hệ thống điện. Dự báo được yêu cầu trên các quy mô địa lý và thời gian khác nhau, có thể được mô hình hóa dưới dạng phân cấp.

Từ đó ta thấy tại nước ngoài có những sự khác biệt về bài toán dự báo:

- *Phát triển mạnh:* Tại các quốc gia phát triển, bài toán dự báo đã được phát triển mạnh và có sự ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, thương mại điện tử, y tế, và năng lượng.
- *Sự kết hợp của công nghệ mới:* Các quốc gia nước ngoài thường kết hợp sự phát triển của công nghệ mới như trí tuệ nhân tạo, học máy và big data analytics để cải thiện hiệu suất của bài toán dự báo.
- *Tổng hợp dữ liệu:* Một ưu điểm của các quốc gia phát triển là có khả năng tổng hợp dữ liệu từ nhiều nguồn khác nhau, tạo nền tảng cho việc dự báo chính xác hơn và đa dạng hơn.

## 1.4. Bài toán phân tích mô tả thị trường vàng và dự báo giá vàng bằng mô hình hồi quy

Bài toán "Phân tích mô tả thị trường vàng và dự báo giá vàng bằng mô hình hồi quy" là một nghiên cứu trong lĩnh vực phân tích dữ liệu, tập trung vào việc hiểu và dự đoán giá vàng trên thị trường tài chính. Mục tiêu của bài toán là phân tích giá vàng và các yếu tố ảnh hưởng đến giá vàng, sử dụng phương pháp hồi quy để xây dựng một mô hình dự báo hiệu quả.

- ***Mục tiêu nghiên cứu:***
  + *Phân tích yếu tố ảnh hưởng:* Hiểu rõ các yếu tố có thể tác động đến giá vàng, chẳng hạn như tỷ giá hối đoái của đồng đô la Mỹ, lãi suất, giá dầu, và các chỉ số kinh tế khác. Các yếu tố này đóng vai trò quan trọng trong việc xác định xu hướng biến động của giá vàng.
  + *Xây dựng mô hình hồi quy:* Sử dụng phương pháp hồi quy để xây dựng mô hình dự báo giá vàng dựa trên các yếu tố ảnh hưởng đã được xác định. Mô hình hồi quy sẽ cố gắng tìm ra mối quan hệ giữa các biến độc lập (các yếu tố ảnh hưởng) và biến phụ thuộc (giá vàng).

+ *Dự đoán giá vàng:* Dựa trên mô hình hồi quy đã xây dựng, mục tiêu là dự đoán giá vàng và xu hướng giá vàng, dựa trên các thông tin về các yếu tố ảnh hưởng đã biết.

- ***Ý nghĩa khoa học và thực tiễn:***

+ *Thị trường tài chính:* Đề tài này đóng góp vào lĩnh vực tài chính và đầu tư bằng cách áp dụng các kỹ thuật phân tích dữ liệu và hồi quy để khám phá mối liên hệ giữa các yếu tố kinh tế và giá vàng, từ đó cung cấp thông tin hữu ích cho việc ra quyết định đầu tư.

+ *Quản lý rủi ro:* Kết quả của nghiên cứu có thể giúp các nhà đầu tư và tổ chức tài chính hiểu rõ hơn về các yếu tố ảnh hưởng đến giá vàng, hỗ trợ trong việc quản lý rủi ro và tối ưu hóa danh mục đầu tư.

+ *Tư duy phân tích:* Việc thực hiện phân tích dữ liệu và xây dựng mô hình hồi quy trong ngữ cảnh này cũng giúp phát triển kỹ năng tư duy phân tích và khả năng áp dụng các phương pháp phân tích vào các vấn đề thực tế trong lĩnh vực tài chính.

Như vậy, bài toán này không chỉ mang ý nghĩa đối với thị trường vàng mà còn cung cấp những kiến thức hữu ích cho việc phân tích và dự báo trong các lĩnh vực tài chính và kinh tế.

## 1.5. Kết luận chương 1

Chương 1 đã trình bày tổng quan về đề tài, bao gồm việc giới thiệu về phân tích dữ liệu và bài toán dự báo, cũng như các khái niệm cơ bản liên quan đến phân tích dữ liệu trong thị trường tài chính. Chương này cũng đã mô tả tình hình nghiên cứu trong và ngoài nước về dự báo giá vàng, đồng thời trình bày chi tiết về bài toán phân tích mô tả thị trường vàng và dự báo giá vàng bằng phương pháp hồi quy. Những nội dung này sẽ làm nền tảng cho các chương tiếp theo trong việc áp dụng các phương pháp kỹ thuật để phân tích và dự báo.

# CHƯƠNG 2. MỘT SỐ PHƯƠNG PHÁP

## 2.1. Phương pháp phân tích mô tả

### *2.1.1. Phương pháp phân tích trên từng biến*

Khi thực hiện phân tích trên một biến (hoặc một thuộc tính), mục tiêu chính là hiểu rõ các đặc điểm cơ bản của biến đó. Điều này thường bao gồm xác định và xử lý các giá trị ngoại lai hoặc bất thường (Outliers). Đây là các giá trị dữ liệu mà rất khác biệt so với phần lớn các giá trị khác trong tập dữ liệu. Các giá trị ngoại lai có thể xuất hiện do lỗi nhập liệu, lỗi đo lường, hoặc đơn giản là do các sự kiện hiếm gặp.

Việc xác định các Outliers có vai trò quan trọng và là mắt xích liên kết giữa phân tích mô tả và phân tích hồi quy, bởi vì ta có thể tiến hành làm sạch những giá trị này tại công đoạn tiền xử lý dữ liệu của phân tích hồi quy. Cụ thể với từng loại dữ liệu khác nhau, ta sẽ phân tích như sau:

- ***Dữ liệu số:***
  - *Biểu đồ Histogram:* Biểu đồ hiển thị tần suất xuất hiện của các khoảng giá trị dữ liệu.
  - *Các đại lượng thống kê:* Bao gồm mean (trung bình), stdev (độ lệch chuẩn), median (trung vị), quartile (phân vị) ... Các giá trị này giúp mô tả trung bình, phương sai và phân phối của dữ liệu.
  - *Biểu đồ Box & Whisker (Boxplot):* Biểu đồ hiển thị tổng quan giá trị đó bao gồm các giá trị đại lượng thống kê đã tính được.
- ***Dữ liệu phi số:***
  - *Bảng tần suất (Frequency table):* Biểu đồ liệt kê các giá trị khác nhau của biến và số lần xuất hiện của mỗi giá trị.
  - *Biểu đồ cột (Bar chart):* Biểu đồ thể hiện tần suất của từng giá trị dữ liệu dưới dạng các cột đứng.
  - *Biểu đồ hình tròn hoặc donut (Pie chart, Donut chart):* Biểu đồ thể hiện phần trăm tần suất của từng giá trị trong tổng số.

### *2.1.2. Phương pháp phân tích trên nhiều biến*

Phân tích trên nhiều biến hướng tới việc hiểu mối quan hệ và tương tác giữa các biến trong tập dữ liệu. Điều này có thể giúp bạn phát hiện ra các mẫu, xu hướng hoặc tương quan có thể tồn tại giữa chúng.

Các mối liên hệ giữa các biến (Interrelationships) có thể là nhiều dạng khác nhau: Mối tương quan tuyến tính, tương quan không tuyến tính, tương quan ngược... Với mỗi mối liên hệ, ta có thể phân tích và tìm ra được cách các biến tương tác và ảnh hưởng lẫn nhau.

Việc phân tích trên nhiều biến cũng có mối liên hệ mật thiết đến phân tích hồi quy khi giúp ta xác định được các giá trị ngoại lai của dữ liệu. Do là phân tích nhiều biến, vậy nên sẽ có 3 kiểu dữ liệu phân tích khác nhau: số, phi số và hỗn hợp (cả số và phi số):

- ***Dữ liệu số:***

+ *Scatter Plot (Biểu đồ Scatter):* Biểu đồ thể hiện mối quan hệ giữa hai biến số. Mỗi điểm trên biểu đồ thể hiện một cặp giá trị của hai biến trên trục ngang và dọc. Biểu đồ này dùng để tìm kiếm sự tương quan giữa 2 biến số như tương quan tuyến tính hoặc không tuyến tính.

+ *Bảng dữ liệu thống kê (Statistical Summary Table):* Tạo bảng để liệt kê các đại lượng thống kê (mean, median, stdev…) giữa các biến số của dữ liệu.

- ***Dữ liệu phi số:***

+ *Bảng dữ liệu thống kê (Statistical Summary Table):* Cũng là bảng dữ liệu thống kê nhưng với giá trị phi số, đó sẽ chỉ có giá trị tần suất xuất hiện (mode) của dữ liệu.

- ***Dữ liệu hỗn hợp:***

+ *Bảng thống kê tổng hợp:* Đây là sự kết hợp giữa bảng dữ liệu thống kê của dữ liệu số và phi số. Sự kết hợp tổng quan này sẽ cho ta bao quát được phân bổ của dữ liệu.

+ *Biểu đồ Box & Whisker (Boxplot**):*** Được sử dụng để so sánh phân phối của một dữ liệu số với tần suất của một dữ liệu phi số. Biểu đồ

này sẽ cho ta mối quan hệ mật thiết về sự ảnh hưởng của các giá trị phi số lên giá trị số được phân tích.

## 2.2. Phương pháp phân tích hồi quy

### *2.2.1. Tổng quan về phân tích hồi quy*

Phân tích hồi quy là một tập hợp các phương pháp thống kê được sử dụng để ước tính các mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Nó có thể được sử dụng để đánh giá sức mạnh của mối quan hệ giữa các biến và để mô hình hóa mối quan hệ trong tương lai giữa chúng.

Phân tích hồi quy là một cách phân loại toán học để xác định biến nào trong số những biến đó thực sự có tác động. Nó trả lời các câu hỏi: Yếu tố nào quan trọng nhất? Cái nào có thể bỏ qua? Các yếu tố đó tương tác với nhau như thế nào? Và quan trọng nhất, chúng ta chắc chắn như thế nào về tất cả những yếu tố này?

Trong phân tích hồi quy, ta cần xác định một biến phụ thuộc – yếu tố chính mà ta đang cố gắng hiểu hoặc dự đoán. Phân tích hồi quy bao gồm một số biến thể, chẳng hạn như tuyến tính, nhiều tuyến tính và phi tuyến tính. Trong đó mô hình phổ biến là tuyến tính và nhiều tuyến tính. Đối với phân tích hồi quy phi tuyến, chúng thường được sử dụng cho các tập dữ liệu phức tạp hơn trong đó các biến phụ thuộc và độc lập thể hiện mối quan hệ phi tuyến.

### *2.2.2. Các phương pháp phân tích hồi quy*

Để phân tích hồi quy có rất nhiều phương pháp để phân tích. Dưới đây sẽ là một số phương pháp quan trọng dùng để phân tích hồi quy:

*Hồi quy tuyến tính (Linear Regression):* Hồi quy tuyến tính dự đoán giá trị mục tiêu dựa trên biến độc lập bằng cách tìm đường thẳng "tốt nhất" vượt qua dữ liệu. Phương pháp này đơn giản và phù hợp với dữ liệu có mối quan hệ tuyến tính. Tuy nhiên, nó có thể không xử lý được dữ liệu phi tuyến và ảnh hưởng bởi nhiễu dữ liệu.

*Hồi quy Ridge (Ridge Regression):* Hồi quy Ridge là phiên bản cải tiến của hồi quy tuyến tính bằng cách thêm hệ số điều chuẩn l2 vào hàm mất mát. Điều này giúp kiểm soát độ phức tạp của mô hình và tránh tình trạng quá khớp

(overfitting). Tuy ưu điểm là giảm overfitting và xử lý đa cộng tuyến, nhưng cần lựa chọn tham số điều chuẩn chính xác.

*Hồi quy Lasso (Lasso Regression):* Hồi quy Lasso cũng cải tiến từ hồi quy tuyến tính, nhưng thay vì l2, nó sử dụng hệ số điều chuẩn l1 để thúc đẩy một số hệ số về 0. Điều này dẫn đến lựa chọn biến tự động và giảm  biến quan trọng. Lasso giải quyết vấn đề "chọn biến" nhưng cần phải có tham số điều chuẩn chính xác.

*Hồi quy mạng nơ-ron nhân tạo (Neural Network Regression):* Đây là một phương pháp học máy sử dụng các mạng nơ-ron để giải quyết các bài toán hồi quy, tức là dự đoán một giá trị liên tục dựa trên các đầu vào. Mạng nơ-ron nhân tạo có khả năng mô hình hóa các mối quan hệ phi tuyến tính phức tạp giữa các biến độc lập và biến phụ thuộc, điều này làm cho nó trở thành một công cụ mạnh mẽ trong việc giải quyết các bài toán hồi quy phức tạp mà các mô hình hồi quy truyền thống (như hồi quy tuyến tính) không thể xử lý hiệu quả. Một số mạng nơ-ron dùng trong hồi quy có thể kể đến như mạng nơ-ron truyền thẳng (Feedforward Neural Networks - FNN), mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) và Mạng nơ-ron LSTM (Long Short-Term Memory).

### *2.2.3. Lựa chọn phương pháp*

LSTM (Long Short-Term Memory) là một loại mạng nơ-ron được thiết kế đặc biệt để xử lý và dự báo các chuỗi thời gian (time series) hoặc dữ liệu có phụ thuộc theo thứ tự (sequential data). Phương pháp này sẽ giúp ta đạt được mục tiêu nghiên cứu và trả lời những câu hỏi quan trọng. Vì vậy chúng  em lựa chọn phương pháp mạng nơ-ron LSTM để thực hiện thực nghiệm.

## 2.3. Công cụ phục vụ thực hiện bài toán

### *2.3.1. Python*

*Hình 2.1. Ngôn ngữ lập trình Python [2]*

Python là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, thường được sử dụng để xây dựng trang web và phần mềm, tự động hoá các tác vụ và tiến hành phân tích dữ liệu. Với sự phát triển của khoa học dữ liệu hiện nay, Python lại càng được ứng dụng rộng rãi hơn trong ngành Data Analyst. Với thư viện đa dạng trong các lĩnh vực như khai thác dữ liệu (Scrapy, BeautifulSoup4, …), xử lý dữ liệu và mô hình hóa (Pandas, Scikit-learn, …), trực quan hóa dữ liệu (Matplotlib, Plotly, …) thì đây là một lựa chọn tuyệt vời để phân tích dữ liệu. Tuy nhiên bên cạnh những ưu điểm về thư viện cũng như cộng đồng lập trình đông đảo, Python vẫn vướng phải một số nhược điểm, đó là bị giới hạn tốc độ, mức tiêu thụ bộ nhớ cao và không phải là một ngôn ngữ được hỗ trợ nhiều cho môi trường di động.

## 2.4. Kết luận chương 2

Chương 2 đã trình bày các phương pháp kỹ thuật, cụ thể là phương pháp phân tích mô tả, phương pháp phân tích hồi quy và các công cụ thực hiện bài toán. Đồng thời quyết định sử dụng mô hình mạng nơ-ron LSTM và ngôn ngữ Python để thực hiện thực nghiệm.

# CHƯƠNG 3. THỰC NGHIỆM

## 3.1. Dữ liệu thực nghiệm

Trong project này, bộ dữ liệu được phân tích ở đây là file dataset (.csv) chứa 3728 thông tin về ngày giao dịch, giá vàng (Close_Gold), giá dầu (Close_Oil), chỉ số S&P500 (SP500) và chỉ số đồng Dollar (DollarIndex).

Cụ thể thông tin như sau:

- Tên bộ dữ liệu: gold_oil_dollarindex_sp500
- Thư viện sử dụng để lấy dữ liệu: Thư viện yfinance trong Python.

(Nguồn: https://pypi.org/project/yfinance)

- Dữ liệu:

https://docs.google.com/spreadsheets/d/1ySFiATu8WMckvlY6-Q4Chw3aNrxeCevUfFRb0mNpUHY/edit?usp=sharing

- Dữ liệu 17 dòng đầu của dataset:

| | Date | Open | High | Low | Close_Gold | Volume | SP500 | Close_Oil | DollarIndex |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1/4/2010 | 1117.7 | 1122.3 | 1097.1 | 1117.7 | 184 | 1132.99 | 81.51 | 77.83 |
| 2 | 1/5/2010 | 1118.1 | 1126.5 | 1115 | 1118.1 | 53 | 1136.52 | 81.77 | 77.85 |
| 3 | 1/6/2010 | 1135.9 | 1139.2 | 1120.7 | 1135.9 | 363 | 1137.14 | 83.18 | 77.65 |
| 4 | 1/7/2010 | 1133.1 | 1133.1 | 1129.2 | 1133.1 | 56 | 1141.69 | 82.66 | 78.11 |
| 5 | 1/8/2010 | 1138.2 | 1138.2 | 1122.7 | 1138.2 | 54 | 1144.98 | 82.75 | 77.65 |
| 6 | 1/11/2010 | 1150.7 | 1161.2 | 1143 | 1150.7 | 177 | 1146.98 | 82.52 | 77.16 |
| 7 | 1/12/2010 | 1128.9 | 1157.2 | 1127.2 | 1128.9 | 51 | 1136.22 | 80.79 | 77.12 |
| 8 | 1/13/2010 | 1136.4 | 1136.4 | 1121 | 1136.4 | 58 | 1145.68 | 79.65 | 76.99 |
| 9 | 1/14/2010 | 1137 | 1145.9 | 1132.8 | 1142.6 | 81 | 1148.46 | 79.39 | 76.89 |
| 10 | 1/15/2010 | 1132.8 | 1133.4 | 1127.2 | 1130.1 | 50 | 1136.03 | 78 | 77.48 |
| 11 | 1/19/2010 | 1139.7 | 1139.7 | 1130.5 | 1139.7 | 22 | 1150.23 | 79.02 | 77.65 |
| 12 | 1/20/2010 | 1123.3 | 1133 | 1109.8 | 1112.3 | 28 | 1138.04 | 77.62 | 78.5 |
| 13 | 1/21/2010 | 1102.7 | 1107.5 | 1090.2 | 1102.7 | 99 | 1116.48 | 76.08 | 78.51 |
| 14 | 1/22/2010 | 1089.2 | 1095.4 | 1083 | 1089.2 | 58 | 1091.76 | 74.54 | 78.43 |
| 15 | 1/25/2010 | 1095.2 | 1095.2 | 1095.2 | 1095.2 | 8 | 1096.78 | 75.26 | 78.35 |
| 16 | 1/26/2010 | 1097.9 | 1097.9 | 1097.9 | 1097.9 | 4 | 1092.17 | 74.71 | 78.61 |
| 17 | 1/27/2010 | 1084.4 | 1084.4 | 1084.4 | 1084.4 | 206694 | 1097.5 | 73.67 | 78.87 |

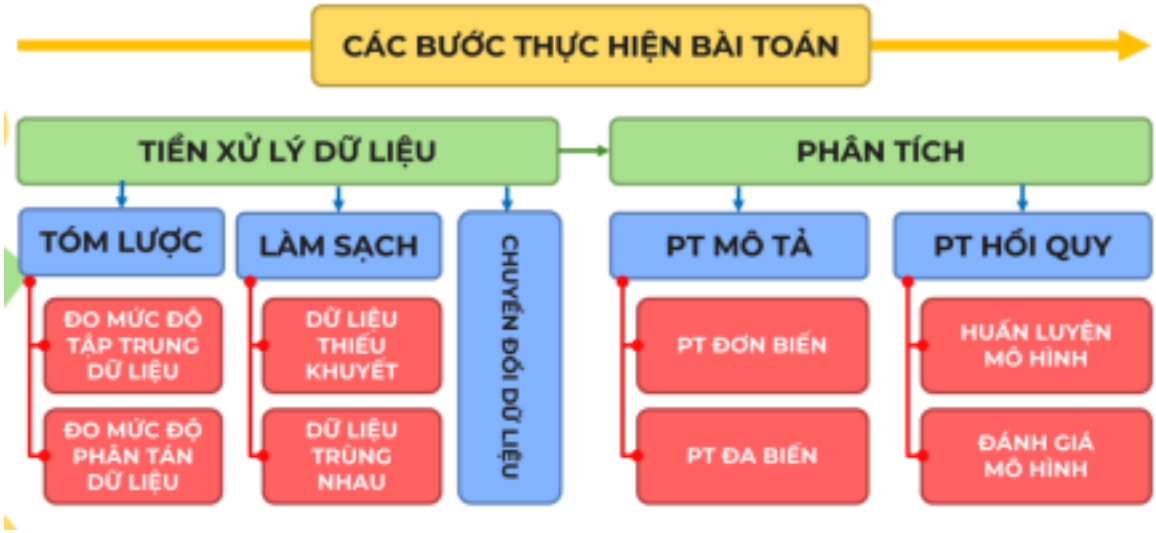
*Hình 3.1. 17 dòng đầu của bộ dữ liệu gốc*

- Thông tin cụ thể các cột của dataset như sau:

+ *"Date":* Ngày của phiên giao dịch.
+ *"Open":* Giá mở phiên giao dịch của vàng.
+ *"High":* Giá cao nhất trong phiên giao dịch của vàng.
+ *"Low":* Giá thấp nhất trong phiên giao dịch của vàng.
+ *"Close_Gold":* Giá đóng phiên giao dịch của vàng.

+ *"Volume":* Khối lượng giao dịch trong phiên của vàng.
+ *"SP500":* Chỉ số đóng phiên giao dịch của chỉ số S&P500 (chỉ số đại diện cho thị trường chứng khoán Mỹ).
+ *"Close_Oil":* Giá đóng phiên giao dịch của Dầu.
+ *"DollarIndex":* Chỉ số đóng phiên giao dịch của đồng DollarIndex.

## 3.2. Quy trình thực nghiệm




*Hình 3.2. Quy trình thực nghiệm đề tài phân tích dữ liệu*

### *3.2.1. Đặt mục tiêu*

- Mục đích của phân tích mô tả:

Phân tích mô tả giúp tóm tắt dữ liệu thị trường vàng, bao gồm các yếu tố như giá mở cửa, giá đóng cửa, khối lượng giao dịch, cùng với biến động của chỉ số SP500, giá dầu và chỉ số Dollar Index. Mục tiêu là nhận diện xu hướng giá vàng qua các giai đoạn lịch sử và phát hiện những yếu tố chính ảnh hưởng đến sự biến động của giá.

- Mục đích của phân tích hồi quy:

Phân tích hồi quy nhằm xây dựng mô hình dự đoán giá vàng dựa trên các biến độc lập như giá dầu, chỉ số SP500 và chỉ số Dollar Index. Mô hình sẽ giúp xác định mối quan hệ giữa các yếu tố này với giá vàng, từ đó hỗ trợ đưa ra dự báo cho các quyết định đầu tư.

### *3.2.2. Tiền xử lý dữ liệu*

#### *a. Tóm lược dữ liệu*

Tóm lược dữ liệu trong phân tích dữ liệu là quá trình tổng hợp, trích xuất và trình bày các thông tin quan trọng và chính xác từ tập dữ liệu ban đầu. Mục tiêu của việc tóm lược dữ liệu là giúp người đọc hoặc người xem nắm bắt được những điểm quan trọng và khái quát của dữ liệu mà không cần phải đọc hoặc xem toàn bộ dữ liệu gốc.

- Tóm lược dữ liệu bao gồm 2 loại đo:

+ Đo mức độ tập trung dữ liệu (mean, median, mode,…).
+ Đo mức độ phân tán dữ liệu (Q1, Q2, Q3, IQR, standard deviation).

Ta sẽ tiến hành tổng hợp các thông tin về độ tập trung và phân tán của dữ liệu. Những thông số này chỉ tương thích với các cột dữ liệu dạng thông số, vậy nên sẽ chỉ có các cột 'Open', 'High', 'Low', 'Close_Gold', 'Volume', 'Close_SP500', 'Close_Oil', 'DollarIndex' là được phân tích.

Dưới đây là kết quả tóm lược dữ liệu bao gồm các thuộc tính count, mean, median, stdev, min, Q1, Q2,Q3, max, mode ,variance,IQR của các dữ liệu trên:

| | <anonymous> | Open | High | Low | Close_Gold | Volume | SP500 | Close_Oil | DollarIndex |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Count | 3727.0 | 3727.0 | 3727.0 | 3727.0 | 3727.0 | 3727.0 | 3727.0 | 3727.0 |
| 2 | Mean | 1531.55 | 1539.15 | 1523.61 | 1531.51 | 5355.95 | 2689.61 | 72.03 | 92.01 |
| 3 | Median | 1423.3 | 1430.4 | 1416.0 | 1422.6 | 157.0 | 2404.39 | 73.44 | 94.28 |
| 4 | Mode | 1252.7 | 1236.0 | 1218.6 | 1273.7 | 0.0 | 1064.88 | 44.66 | 80.25 |
| 5 | Min | 1052.2 | 1062.0 | 1045.2 | 1050.8 | 0.0 | 1022.58 | -37.63 | 73.11 |
| 6 | Max | 2748.6 | 2760.9 | 2729.1 | 2760.8 | 386334.0 | 5864.67 | 123.7 | 114.05 |
| 7 | Q1 | 1258.4 | 1265.15 | 1253.35 | 1258.25 | 44.5 | 1692.66 | 53.19 | 82.1 |
| 8 | Q2 | 1423.3 | 1430.4 | 1416.0 | 1422.6 | 157.0 | 2404.39 | 73.44 | 94.28 |
| 9 | Q3 | 1778.85 | 1788.5 | 1770.4 | 1777.9 | 504.0 | 3751.76 | 89.88 | 98.43 |
| 10 | IQR | 520.45 | 523.35 | 517.05 | 519.65 | 459.5 | 2059.1 | 36.68 | 16.34 |
| 11 | Variance | 109649.67 | 110923.09 | 108619.82 | 109795.41 | 829620855.76 | 1492800.04 | 461.47 | 84.99 |
| 12 | Stdev | 331.13 | 333.05 | 329.58 | 331.35 | 28803.14 | 1221.8 | 21.48 | 9.22 |

*Hình 3.3. Thông tin tóm lược dữ liệu của cột dữ liệu dạng số*

Trong đó:

+ *Count:* Cho biết tổng số bản ghi trong tập dữ liệu, giúp ta nắm được quy mô của dữ liệu. Số lượng lớn có thể làm cho các kết quả thống kê trở nên đáng tin cậy hơn.
+ *Median:* Cung cấp thông tin về giá trị ở giữa trong tập dữ liệu.

+ *Mean:* Cho biết giá trị trung bình của giá vàng, giúp xác định xu hướng chung.
+ *Độ Lệch Chuẩn - std (Standard Deviation):* Đo lường mức độ phân tán của dữ liệu xung quanh giá trị trung bình. Độ lệch chuẩn cao cho thấy giá vàng có sự biến động lớn, trong khi độ lệch chuẩn thấp cho thấy sự ổn định hơn.
+ *Các tứ phân vị (Q1 - 25%, Q2 – 50%, Q3 – 75%):* Giúp mô tả sự phân bố của dữ liệu và xác định mức độ tập trung hay sự phân tán của nó.
+ *IQR (IQR = Q3 – Q1):* Thể hiện sự phân bố của 50% dữ liệu trung tâm, giúp nhận diện giá trị ngoại lệ. IQR hữu ích để đánh giá sự biến động và phân tích độ phân tán trong giá vàng.
+ *Min và Max:* Giúp xác định khoảng biến động của dữ liệu. Biết giá trị tối thiểu và tối đa giúp hiểu rõ giới hạn của giá vàng trong khoảng thời gian phân tích.
+ *Mode:* Thể hiện giá trị phổ biến nhất trong dữ liệu. Mode có thể giúp xác định các mức giá phổ biến mà nhà đầu tư thường giao dịch.

- *Từ bảng thống kê mô tả dữ liệu ta có thể thấy:*

+ Giá vàng biến động khá lớn: Khoảng dao động giữa giá trị tối thiểu (1050.8) và giá trị tối đa (2760.8) là rất lớn, cho thấy giá vàng có sự biến động mạnh trong khoảng thời gian được quan sát.
+ Giá vàng có sự phân bố lệch phải: Giá trị trung bình (1531.51) lớn hơn giá trị trung vị (1422.6), cho thấy dữ liệu có xu hướng lệch phải. Điều này có thể do một số giá trị rất cao kéo trung bình lên.
+ Biến động cao: Độ lệch chuẩn (331.35) và phương sai (109795.41) lớn cho thấy sự biến động giá vàng rất cao, điều này phản ánh rằng giá vàng thay đổi đáng kể theo thời gian.
+ Tập trung dữ liệu: Khoảng tứ phân vị (Q3 - Q1 = 519.65) cho thấy rằng 50% các giá trị giá vàng nằm trong khoảng từ 1258.25 đến 1777.9. Điều này cung cấp một bức tranh rõ ràng về mức độ tập trung của giá vàng.

***b. Làm sạch dữ liệu***

Làm sạch dữ liệu là quá trình loại bỏ các sai sót, lỗi, nhiễu và thông tin không chính xác hoặc không cần thiết khỏi tập dữ liệu ban đầu để đảm bảo dữ liệu đáng tin cậy và phù hợp cho việc phân tích và xử lý tiếp theo. Quá trình làm sạch dữ liệu thường là một phần quan trọng trong tiền xử lý dữ liệu trước khi bắt đầu phân tích mô tả và cả phân tích hồi quy.

Một số tác vụ chính trong quá trình làm sạch dữ liệu bao gồm:

+ *Loại bỏ dữ liệu trùng lặp:* Loại bỏ các bản ghi bị trùng lặp trong tập dữ liệu để tránh ảnh hưởng đến kết quả phân tích.
+ *Xử lý dữ liệu thiếu:* Điền vào các giá trị thiếu hoặc quyết định loại bỏ chúng dựa trên ngữ cảnh và mục tiêu của phân tích.
+ *Sửa lỗi và sai sót:* Điều tra và sửa các lỗi cú pháp, sai sót chính tả hoặc sai sót logic trong dữ liệu.
+ *Chọn lọc đặc trưng:* Xác định và lựa chọn các đặc trưng quan trọng nhất để sử dụng trong phân tích hoặc mô hình hóa.

Đối với project hiện tại, sau khi khảo sát chi tiết các cột dữ liệu, việc sửa lỗi sai sót và chọn lọc đặc trưng cho dataset không quá quan trọng nên ta sẽ tiến hành loại bỏ dữ liệu trùng lặp và xử lý dữ liệu thiếu. Để làm điều này, trước hết ta cần khảo sát số data bị thiếu và trùng lặp. Phương thức “isnull” được sử dụng để kiểm tra các giá trị bị thiếu (hoặc null) trong DataFrame và “duplicated” được sử dụng để xác định các hàng dữ liệu trùng lặp trong DataFrame:

```
[ ]  def missing_data(df):
       data_na = (df.isnull().sum() / len(df)) * 100
       missing_data = pd.DataFrame({ 'Ty le thieu data': data_na })
       print(missing_data)
```

```
[ ]  missing_data(df)
```

```
[ ]  def check_duplicates(df):
       duplicated_rows_data = df.duplicated().sum()
       print(f"\nSO LUONG DATA BI TRUNG LAP: {duplicated_rows_data}")
       data = df.drop_duplicates()
```

```
[ ]  check_duplicates(df)
```

Kết quả khảo sát như sau:

```
                Ty le thieu data
Date                         0.0
Open                         0.0
High                         0.0
Low                          0.0
Close_Gold                   0.0
Volume                       0.0
SP500                        0.0
Close_Oil                    0.0
DollarIndex                  0.0

SO LUONG DATA BI TRUNG LAP: 0
```

*Hình 3.4. Thông tin tỷ lệ thiếu, hụt của dữ liệu và tổng số dữ liệu trùng lặp*

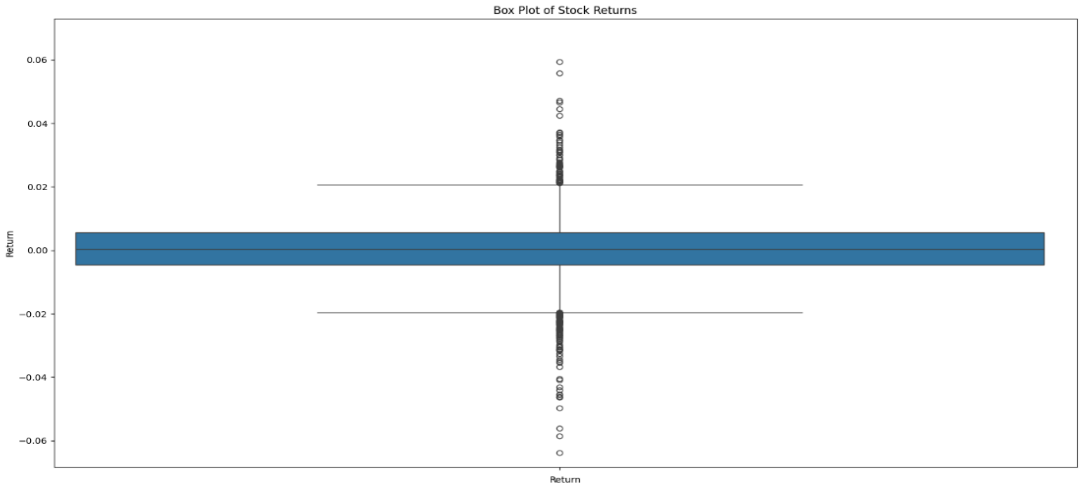
Qua khảo sát, ta đánh giá được tài liệu không có vùng bị thiếu, dữ liệu bị không có hàng nào bị trùng lặp.

### *3.2.3. Phân tích mô tả*

Phân tích mô tả trong phân tích dữ liệu là quá trình tóm tắt, mô tả và hiểu sâu về các đặc điểm, mẫu thái và thông tin quan trọng của tập dữ liệu. Với mục tiêu đó, ta sẽ tiến hành phân tích mô tả cho bộ dữ liệu của project theo cả 2 hướng phân tích đơn biến (trên từng biến) và phân tích đa biến (trên nhiều biến) bằng cách biểu diễn dưới các biểu đồ khác nhau.

### *a. Phân tích đơn biến*

- **Biểu đồ 1: Biểu đồ hộp (Box & whisker)**
  - Dạng biểu đồ: Biểu đồ hộp (Box & whisker)
  - Loại phân tích: Đơn biến (‘Daily_returns’)
  - Kiểu dữ liệu: Số thực (float64)




*Hình 3.5. Biểu đồ hộp của lợi nhuận đầu tư (daily_returns)*

- *Mục đích:*

Nhận diện các giá trị daily-return nằm ngoài khoảng phần lớn dữ liệu, từ đó phát hiện các ngày có mức biến động vượt trội, hoặc những bất thường trong lợi nhuận hàng ngày.

- *Công thức tính:*

Bằng cách trừ giá đóng cửa của một tài sản trong một ngày với giá đóng cửa của ngày hôm trước, sau đó chia kết quả cho giá đóng cửa của ngày hôm trước.

- *Nhận xét:*

Phần hộp của biểu đồ rất hẹp, cho thấy rằng 50% lợi nhuận hàng ngày tập trung rất gần với giá trị trung vị. Điều này có nghĩa là phần lớn các ngày đều có lợi nhuận dao động trong một phạm vi hẹp xung quanh mức trung bình, phản ánh một mức biến động nhỏ và ổn định cho phần lớn thời gian.

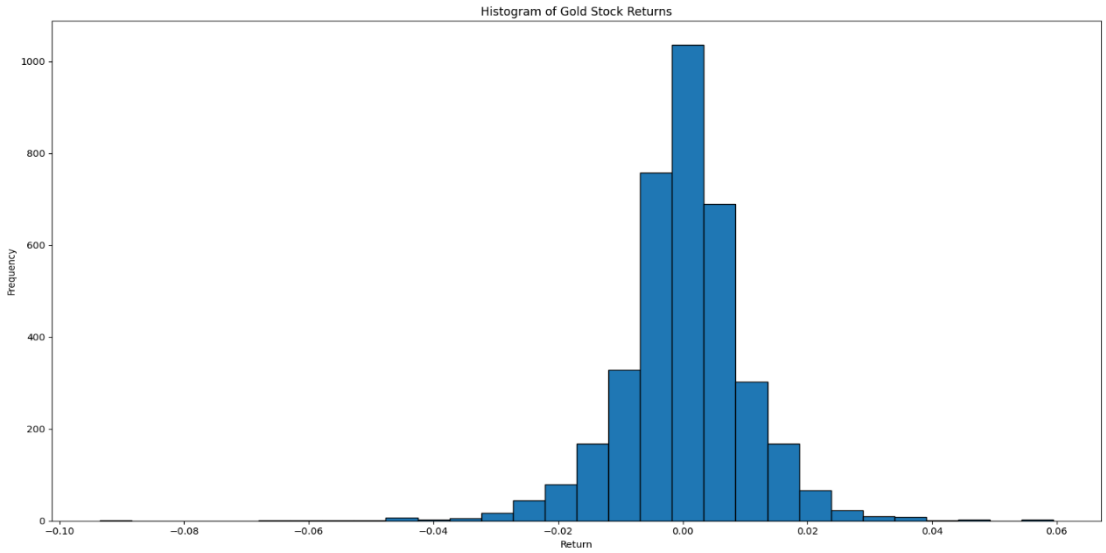
Tuy rằng IQR hẹp, nhưng nó lại có rất nhiều giá trị ngoại lai (outliers) ở cả 2 phía âm dương. Những giá trị này nằm ngoài khoảng 1,5 lần IQR từ Q1 và Q3, biểu thị các ngày có biến động mạnh hơn mức bình thường.

Do có nhiều giá trị ngoại lai (outliers) hơn ở cả hai phía, phân phối của dữ liệu có vẻ không hoàn toàn đồng đều, cho thấy một số ngày có lợi nhuận đột biến so với phần lớn các ngày khác.

- *Kết luận:*

Từ biểu đồ này, cho thấy giá vàng khá ổn định nhưng không hoàn toàn an toàn, vì vẫn có khả năng xuất hiện những ngày biến động mạnh.

- **Biểu đồ 2: Histogram Chart**
  - Dạng biểu đồ: Histogram chart
  - Loại phân tích: Đơn biến (‘Daily_returns’)
  - Kiểu dữ liệu: Số thực (float64)




*Hình 3.6. Biểu đồ Histogram của lợi nhuận đầu tư (daily_returns)*

- *Mục đích:*

Nhận diện các giá trị daily-return nằm ngoài khoảng phần lớn dữ liệu, từ đó phát hiện các ngày có mức biến động vượt trội, hoặc những bất thường trong lợi nhuận hàng ngày.

- *Nhận xét:*

Phân phối gần chuẩn: Lợi nhuận hàng ngày của vàng chủ yếu tập trung quanh mức 0, nghĩa là giá vàng khá ổn định với biến động nhỏ quanh mức trung bình.

Lệch nhẹ về phía âm: Có nhiều ngày lỗ lớn hơn là lãi lớn, cho thấy vàng có thể gặp rủi ro suy giảm nhẹ thường xuyên hơn.

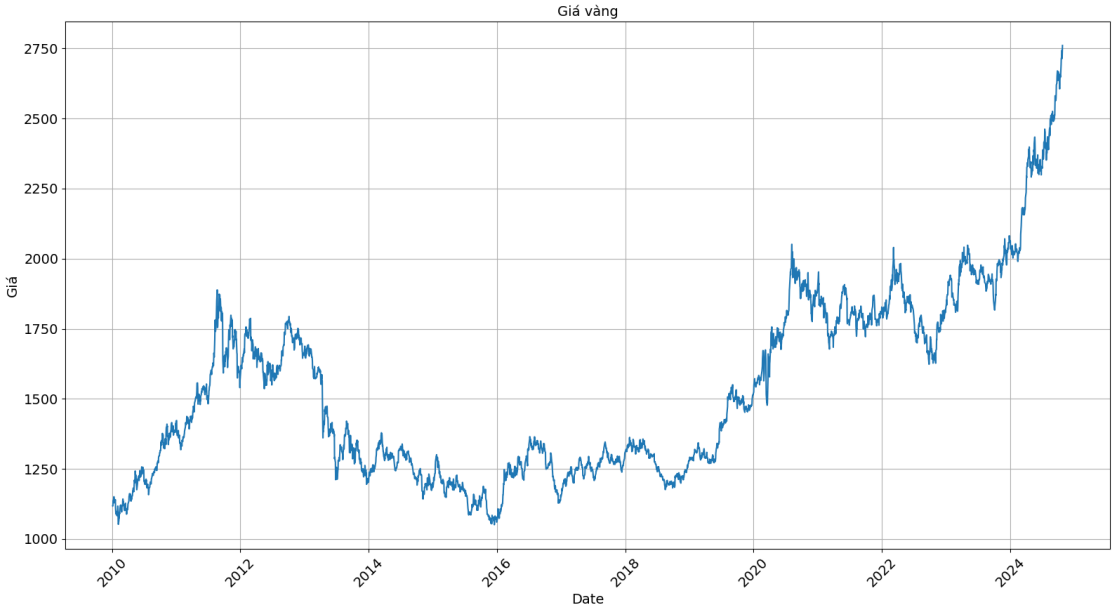
Xuất hiện những giá trị ngoại lai (outliers): Cho thấy rằng một số ngày có lợi nhuận hoặc thua lỗ lớn bất thường, thường do các sự kiện bất ngờ.

- *Kết luận:*

Từ biểu đồ này, vàng là một tài sản có xu hướng ổn định, với phần lớn lợi nhuận dao động gần mức trung bình và ít biến động lớn. Tuy nhiên, sự lệch nhẹ về phía lợi nhuận âm và sự tồn tại của các outliers ở hai phía cho thấy rằng vàng vẫn có rủi ro giảm giá bất ngờ, cần được lưu ý trong chiến lược đầu tư.

***b. Phân tích đa biến***

- **Biểu đồ 3: Biểu đồ đường**
  - Dạng biểu đồ: Hình đường (Line chart)
  - Loại phân tích:
    + Đa biến (‘Date’, ‘Close_Gold’)
    + Đa biến (‘Date’, ‘DollarIndex’ – DXY)
    + Đa biến (‘Date’, ‘Close_Oil’)
    + Đa biến (‘Date’, ‘SP500’)
  - Kiểu dữ liệu: Số thực (float64)



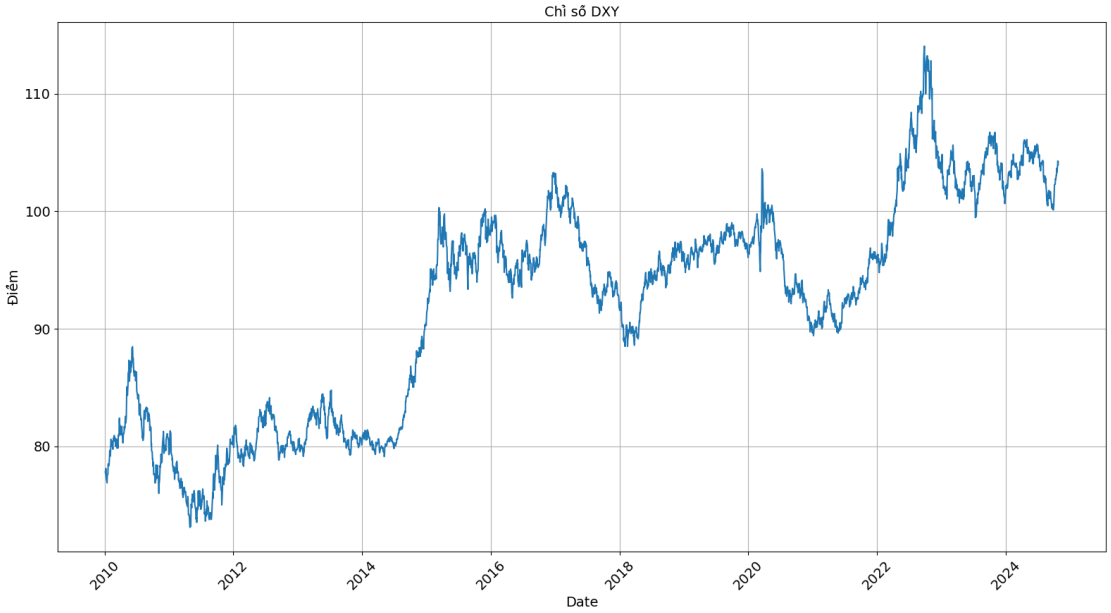

*Hình 3.7. Biểu đồ đường của thuộc tinh "Close_Gold"*

- *Nhận xét biểu đồ đường của thuộc tính "Close_Gold":*

+ Giai đoạn 2010-2014: Giá vàng có sự biến động, dao động trong khoảng từ 1000 đến 1500 USD. Đây là giai đoạn giá vàng khá ổn định, không có sự tăng giảm quá lớn.
+ Giai đoạn 2014-2018: Giá vàng bắt đầu tăng lên, chạm mức 1700 USD vào khoảng năm 2016, sau đó duy trì xu hướng tăng nhẹ.
+ Giai đoạn 2018-2019: Có một số dao động, nhưng giá vẫn duy trì trong khoảng từ 1700 đến 1900 USD.
+ Giai đoạn 2019-2020: Giá vàng tăng mạnh, vượt mức 2000 USD, có thể do sự lo ngại về kinh tế toàn cầu và ảnh hưởng của đại dịch COVID-19.
+ Giai đoạn 2020-2022: Giá vàng tiếp tục tăng, đạt đỉnh mới vào năm 2022 với mức trên 2500 USD.
+ Giai đoạn 2022-2024: Giá vàng dao động mạnh, nhưng vẫn duy trì ở mức cao, khoảng từ 2400 đến 2700 USD.

- *Kết luận:*

Biểu đồ cho thấy sự biến động và xu hướng tăng mạnh của giá vàng trong khoảng thời gian từ 2010 đến 2024, đặc biệt là trong những năm gần đây. Sự tăng giá này có thể phản ánh sự bất ổn của nền kinh tế toàn cầu và nhu cầu đầu tư an toàn vào vàng.




*Hình 3.8. Biểu đồ đường của thuộc tính "DollarIndex" – DXY*

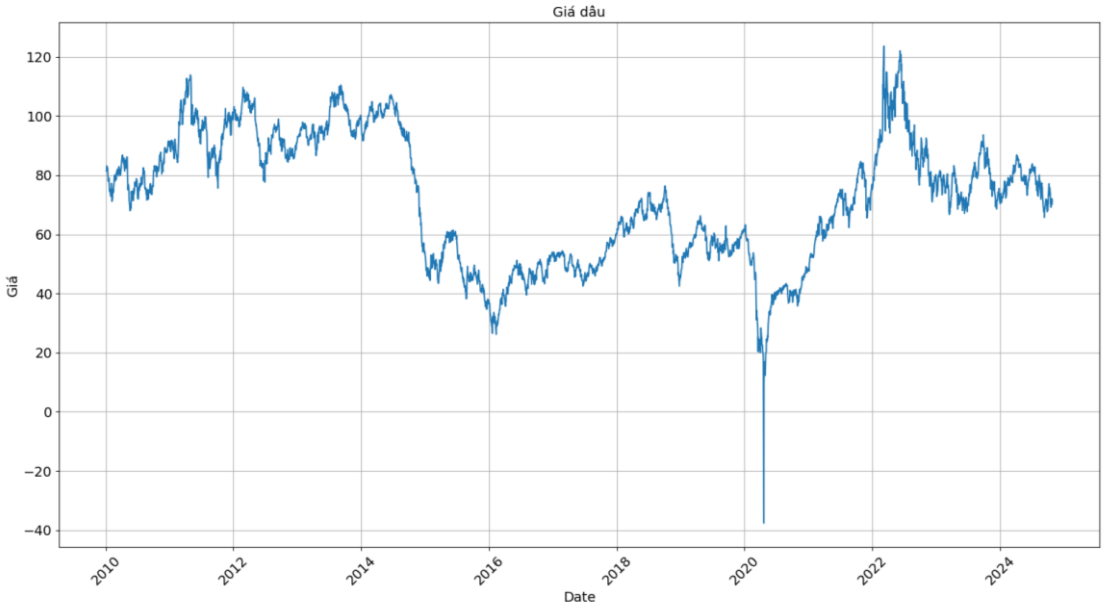
- *Nhận xét biểu đồ đường của thuộc tính "DollarIndex":*

+ Giai đoạn 2010-2014: Chỉ số DollarIndex dao động trong khoảng từ 75 đến 85 điểm, cho thấy sự biến động nhẹ và tương đối ổn định của đồng USD so với các loại tiền tệ khác.
+ Giai đoạn 2014-2016: Chỉ số DollarIndex tăng mạnh, vượt qua ngưỡng 100 điểm vào năm 2015, phản ánh sự mạnh lên của đồng USD.
+ Giai đoạn 2016-2020: Chỉ số có sự biến động mạnh, nhưng nhìn chung vẫn duy trì xu hướng tăng.
+ Năm 2020: Có một sự giảm đột ngột, có thể do tác động của đại dịch COVID-19, nhưng sau đó chỉ số đã hồi phục nhanh chóng.
+ Giai đoạn 2021-2022: Chỉ số DollarIndex tiếp tục tăng mạnh, đạt đỉnh mới vào khoảng 2022.

+ Giai đoạn 2022-2024: Chỉ số có sự giảm nhẹ và duy trì ở mức cao, cho thấy sự ổn định tương đối của đồng USD trong giai đoạn này.

- *Kết luận:*

Biểu đồ cho ta thấy xu hướng chính của chỉ số DollarIndex vẫn là tăng. Giá vàng và giá đô la thường có mối quan hệ ngược chiều. Khi giá trị của đồng đô la Mỹ tăng, giá vàng thường giảm và ngược lại. Điều này là do vàng được giao dịch chủ yếu bằng đô la Mỹ, do đó khi đồng đô la mạnh lên, vàng trở nên đắt hơn đối với các nhà đầu tư sử dụng đồng tiền khác, làm giảm nhu cầu và đẩy giá vàng xuống. Ngược lại, khi đồng đô la yếu đi, vàng trở nên rẻ hơn, dẫn đến tăng cầu và tăng giá.




*Hình 3.9. Biểu đồ đường của thuộc tính "Close_Oil"*

- *Nhận xét biểu đồ đường của thuộc tính "Close_Oil":*

+ Giai đoạn 2010-2014, giá dầu liên tục tăng, đạt đỉnh vào khoảng năm 2012-2013.

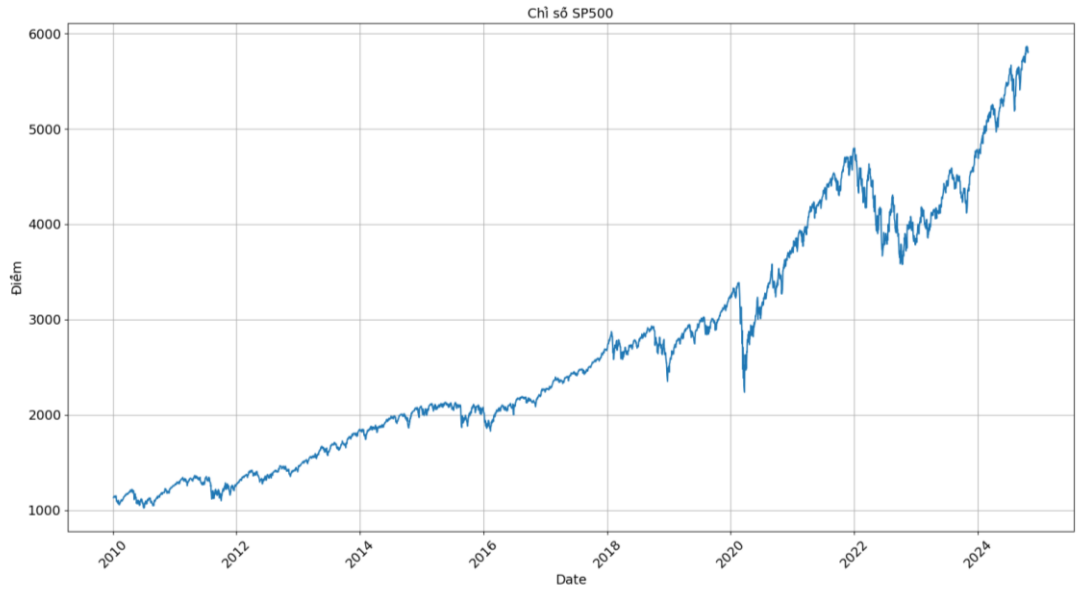
+ Từ năm 2014 đến 2016, giá dầu giảm mạnh, chạm đáy vào năm 2016.

+ Từ năm 2016 đến 2020, giá dầu hồi phục và ổn định hơn, dù vẫn có những dao động nhỏ.

+ Năm 2020, giá dầu giảm đột ngột, có thể do ảnh hưởng của đại dịch COVID-19. Thậm chí, có thời điểm giá dầu đi vào mức âm.
+ Từ năm 2021 đến 2022, giá dầu tăng nhanh, đạt đỉnh mới.
+ Năm 2022 đến 2024, giá dầu có xu hướng giảm với một vài dao động nhỏ.

- *Kết luận:*

Biểu đồ này thể hiện rõ sự biến động của giá dầu qua các năm, những biến động này cho thấy sự không ổn định của thị trường dầu mỏ. Giá dầu có thể ảnh hưởng đến giá vàng thông qua tác động lên nền kinh tế toàn cầu. Khi giá dầu tăng, chi phí sản xuất và vận chuyển cũng tăng, dẫn đến lạm phát cao hơn. Vàng thường được coi là một "kênh trú ẩn an toàn" trong thời kỳ lạm phát, vì vậy giá vàng thường tăng khi giá dầu tăng. Ngược lại, khi giá dầu giảm, lạm phát có thể giảm, dẫn đến sự suy giảm trong nhu cầu đầu tư vào vàng.




*Hình 3.10. Biểu đồ đường của thuộc tinh "SP500"*

- *Nhận xét biểu đồ đường của thuộc tính "SP500":*

+ Giai đoạn 2010-2014: Chỉ số S&P 500 tăng dần, cho thấy sự phát triển ổn định của thị trường chứng khoán trong thời gian này.
+ Giai đoạn 2014-2016: Có một số biến động, nhưng chỉ số vẫn giữ được xu hướng tăng, dù có những khoảng thời gian giảm nhẹ.

+ Giai đoạn 2016-2020: Chỉ số tiếp tục tăng, với những biến động đáng kể nhưng vẫn duy trì được đà tăng.
+ Năm 2020: Chỉ số giảm đột ngột do ảnh hưởng của đại dịch COVID-19, nhưng sau đó đã phục hồi nhanh chóng.
+ Giai đoạn 2021-2022: Chỉ số S&P 500 tăng mạnh, đạt đỉnh mới.
+ Giai đoạn 2022-2024: Có sự dao động nhẹ, nhưng xu hướng chung vẫn là tăng trưởng.

- *Kết luận:*

Biểu đồ đã cho thấy sự tăng trưởng mạnh mẽ và sự biến động của chỉ số S&P 500 trong một khoảng thời gian dài với xu hướng chính vẫn là xu hướng tăng mặc dù có những giai đoạn biến động lớn. Khi thị trường chứng khoán mạnh và chỉ số S&P 500 tăng, nhà đầu tư có xu hướng đầu tư vào cổ phiếu thay vì vàng, vì họ kỳ vọng lợi nhuận cao hơn từ thị trường cổ phiếu. Ngược lại, khi thị trường chứng khoán yếu và chỉ số S&P 500 giảm, nhà đầu tư có xu hướng tìm kiếm các kênh trú ẩn an toàn như vàng, dẫn đến giá vàng tăng.

- **Biểu đồ 4: Biểu đồ nến**
  - Dạng biểu đồ: Nến (Candlestick Charts)
  - Loại phân tích: Đa biến (‘High’,’Low’,’Open’,’Close_Gold’)
  - Kiểu dữ liệu: Số thực (float64)


*Hình 3.11. Biểu đồ nến tính theo USD*

- *Mục đích:*

Biểu đồ được xây dựng để mô tả trực quan dữ liệu giá vàng theo thời gian từ năm 2010 đến năm 2024. Mục tiêu là giúp người đọc dễ dàng nhận diện các xu hướng và sự biến động của giá vàng qua từng giai đoạn, từ đó hỗ trợ việc phân tích các đặc điểm thống kê cơ bản trong bộ dữ liệu.

- *Nhận xét:*

+ *Xu hướng tổng quát:*

Biểu đồ thể hiện một xu hướng tăng trưởng giá vàng theo thời gian, với một số giai đoạn biến động rõ rệt.

Trong dữ liệu, có thể nhận thấy các giai đoạn giá vàng giảm hoặc giữ mức ổn định, xen kẽ với những đợt tăng mạnh.

+ *Biến động dữ liệu:*

Giai đoạn đầu (2010–2013): Giá vàng tăng nhanh và đạt đỉnh, sau đó giảm dần. Đây là giai đoạn biến động mạnh nhất trong dữ liệu.

Giai đoạn giữa (2013–2018): Dữ liệu cho thấy giá vàng dao động trong một phạm vi hẹp hơn, thể hiện sự ổn định tương đối.

Giai đoạn sau (2018–2024): Xu hướng tăng giá trở lại, đặc biệt là từ năm 2020, với mức độ tăng trưởng mạnh hơn so với các giai đoạn trước.

+ *Phân bố dữ liệu:*

Biểu đồ cho thấy sự phân bố giá vàng không đồng đều, với các giai đoạn có độ dốc khác nhau. Điều này thể hiện rằng tốc độ thay đổi của giá vàng không nhất quán và có thể phụ thuộc vào các yếu tố khác trong dữ liệu.

+ *Tầng biểu đồ phụ:*

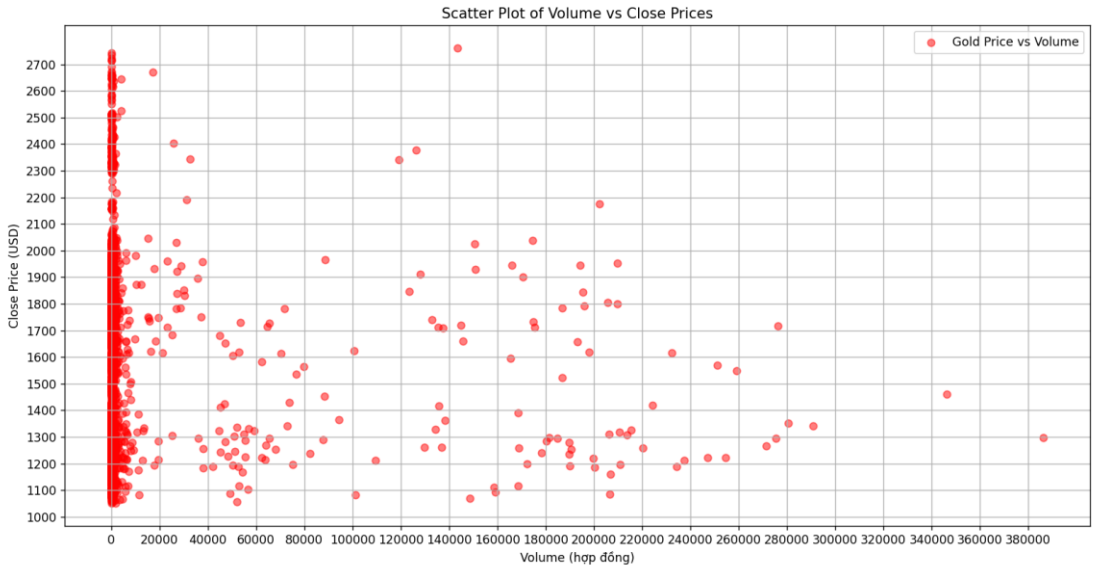
Biểu đồ phụ phía dưới có thể biểu diễn thông tin bổ sung (chẳng hạn khối lượng giao dịch hoặc giá trị trung bình tích lũy theo thời gian). Điều này cung cấp thêm góc nhìn về sự biến thiên của các chỉ số liên quan.

- *Kết luận:*

Biểu đồ minh họa rõ ràng xu hướng và đặc điểm biến động của giá vàng qua từng năm. Với sự phân chia theo thời gian cụ thể, người xem có thể dễ dàng xác định các giai đoạn ổn định, biến động mạnh hoặc tăng trưởng đột biến.

Phân tích trực quan này hỗ trợ bước đầu trong việc xác định các yếu tố ảnh hưởng đến dữ liệu và có thể kết hợp với các kỹ thuật phân tích sâu hơn để làm rõ hơn các mô hình hoặc xu hướng trong dữ liệu.

- **Biểu đồ 5: Biểu đồ phân tán (Scatter Plot)**
  - Dạng biểu đồ: Biểu đồ phân lớp (Scatter Plot)
  - Loại phân tích: Đa biến (‘Close_Gold’ và ‘Volume’)
  - Kiểu dữ liệu: Hỗn hợp (float64 và int64)




*Hình 3.12. Biểu đồ phân tán giữa giá đóng và khối lượng*

- *Mục đích:*

Biểu đồ này cho phép chúng ta quan sát trực quan mối quan hệ giữa Volume (khối lượng giao dịch vàng) và Close Price (giá đóng cửa vàng) trong dữ liệu của bạn.

- *Nhận xét (phân tán dữ liệu):*

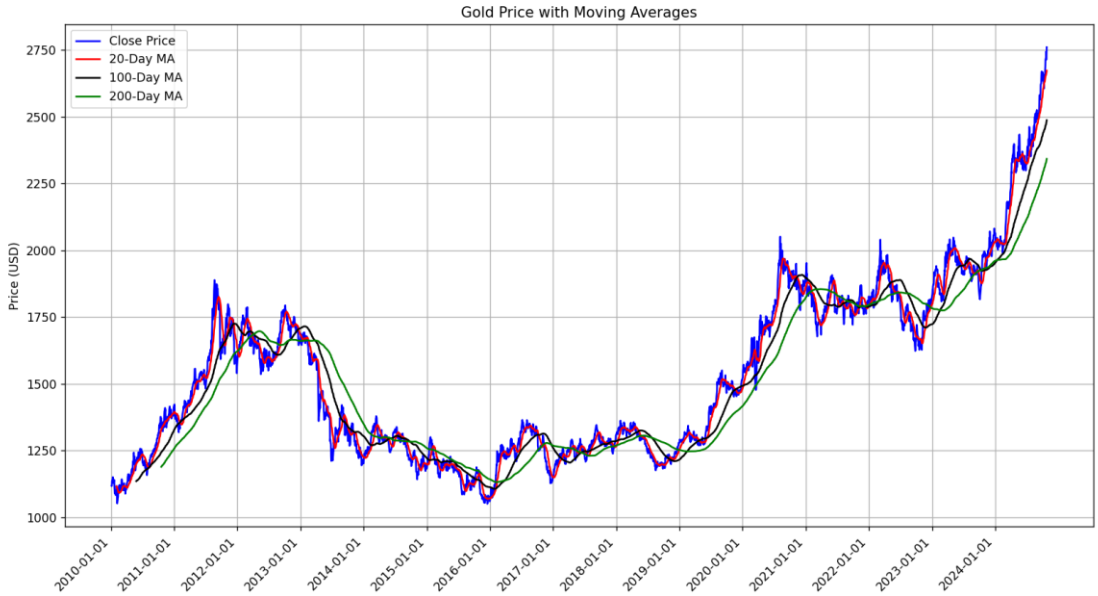
Các điểm đỏ có sự phân tán dày đặc ở phía dưới (gần mốc 0 đến 20000 hợp đồng cho Volume và từ 1050 USD đến 2750 USD cho Close Price). Điều này có nghĩa rằng phần lớn các giao dịch vàng có khối lượng giao dịch nhỏ và giá vàng đóng cửa chủ yếu nằm trong khoảng từ 1100 đến 2100 USD.

Một số điểm phân tán ra xa (lớn hơn 20000 hợp), nhưng giá đóng cửa không thay đổi nhiều so với các giao dịch nhỏ.

- *Kết luận:*

Từ biểu đồ này, có thể rút ra rằng giá vàng không bị ảnh hưởng lớn bởi khối lượng giao dịch, và phần lớn các giao dịch diễn ra với khối lượng tương đối thấp (dưới 20000 hợp đồng).

- **Biểu đồ 6: Biểu đồ trung bình trượt (Moving Average Chart)**
  - Dạng biểu đồ: Biểu đồ trung bình trượt (Moving Average Chart)
  - Loại phân tích: Đa biến (‘Date’ và ‘Close_Gold’)
  - Kiểu dữ liệu: Hỗn hợp (Datetime và float64)




*Hình 3.13. Phân Tích Giá Vàng theo Thời Gian với Trung Bình Trượt*

- *Mục đích:*

Biểu đồ trung bình động (MA) giúp làm mượt các biến động giá vàng hàng ngày, từ đó nhận diện rõ ràng hơn các xu hướng ngắn hạn và dài hạn. Đường MA 20 ngày cung cấp cái nhìn về xu hướng ngắn hạn, trong khi đường MA 100 ngày làm rõ xu hướng trung hạn và MA200 cho thấy xu hướng dài hạn.

- *Nhận xét:*

+ *Đường giá đóng cửa(Close Price):*

Đường màu xanh dương đại diện cho giá đóng cửa của vàng. Đường này có thể thể hiện sự biến động của giá vàng theo thời gian, cho thấy các giai đoạn tăng giá và giảm giá và giai đoạn dao động ít.

+ *Đường trung bình trượt (Moving Averages):*

*MA20 (đường màu đỏ):* Đường trung bình trượt 20 ngày, đường này dao động khá sát với giá đóng cửa, cho thấy xu hướng ngắn hạn của giá vàng. Nếu đường giá (đường màu xanh dương) nằm trên MA20, điều này cho thấy xu hướng tăng trong ngắn hạn, trong khi việc nằm dưới MA20 có thể chỉ ra xu hướng giảm.

*MA100 (đường màu đen):* Đường trung bình trượt 100 so với MA20, MA100 mượt mà hơn và ít dao động, phản ánh xu hướng giá vàng trung hạn. Đường này chậm hơn trong việc phản ứng với các biến động giá, giúp nhận diện xu hướng lâu dài hơn.

*MA200 (đường màu xanh lá):* Đường MA200 giúp xác định xu hướng giá vàng trong dài hạn. Đường MA200 mượt mà nhất trong các đường trung bình động, phản ánh các xu hướng dài hạn trong giá vàng.

+ *Sự tương quan giữa các đường:*

Sự giao cắt giữa đường giá và các đường trung bình trượt có thể mang lại những tín hiệu quan trọng cho các nhà đầu tư:

Khi đường giá cắt lên MA20, MA100 hoặc MA200, có thể xem đây là dấu hiệu của xu hướng tăng, có khả năng tạo ra cơ hội mua vào.

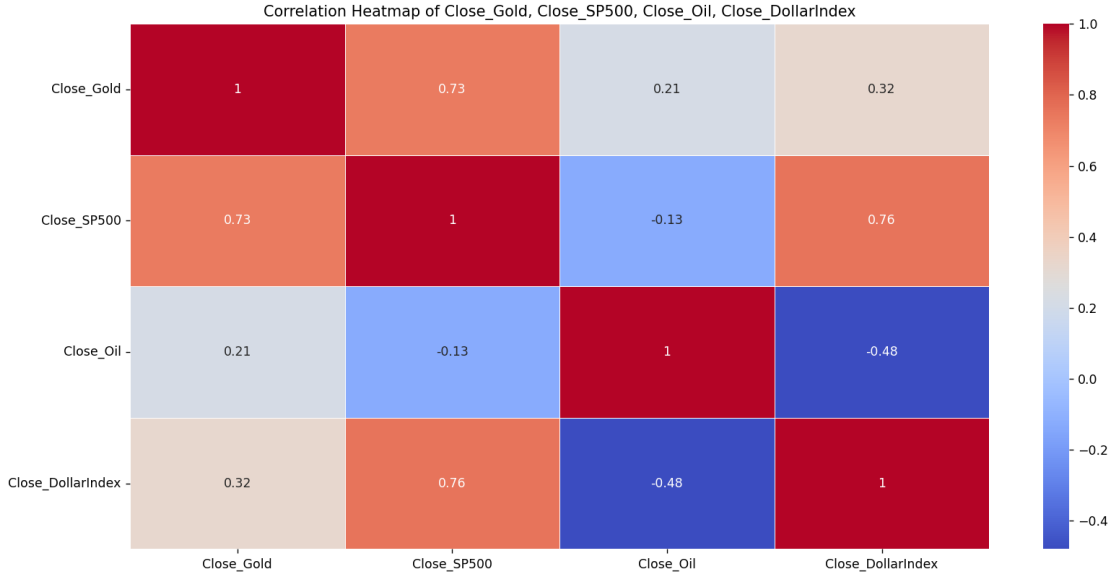
Ngược lại, khi đường giá cắt xuống dưới MA20, MA100 hoặc MA200, điều này có thể báo hiệu xu hướng giảm, có thể dẫn đến quyết định bán ra.

- *Kết luận:*

Biểu đồ này cung cấp một cái nhìn rõ ràng về sự biến động của giá vàng trong suốt thời gian dài với xu hướng tăng mặc dù cũng có những khoảng thời gian điều chỉnh giá. Đường trung bình trượt 20 ngày cho thấy xu hướng ngắn hạn, trong khi đường MA100 phản ánh xu hướng trung hạn và MA200 cho ta

cái nhìn về xu hướng dài hạn. Sự giao cắt giữa đường giá và các đường MA cung cấp tín hiệu quan trọng cho các quyết định đầu tư.

- **Biểu đồ 7: Biểu đồ nhiệt (Heatmap)**
  - Dạng biểu đồ: Heatmap
  - Loại phân tích: Đa biến (Close_Gold, Close_SP500, Close_Oil, Close_DollarIndex)
  - Kiểu dữ liệu: Hỗn hợp (float64)




*Hình 3.14. Biểu đồ nhiệt của các thuộc tính*

- *Mục đích:*

Mục đích của biểu đồ này là để hiển thị mối tương quan giữa các biến số Close_Gold, Close_SP500, Close_Oil và Close_DollarIndex. Biểu đồ này là một ma trận tương quan, trong đó các ô màu đại diện cho giá trị tương quan giữa các cặp biến số. Màu đỏ đậm biểu thị mối tương quan dương mạnh, màu xanh đậm biểu thị mối tương quan âm mạnh, và các màu trung gian biểu thị các mức độ tương quan khác nhau.

- *Nhận xét (Sự tương quan):*

Giá Vàng (Close_Gold) và Chỉ Số S&P 500 (Close_SP500) tương quan mạnh mẽ dương (0.73). Điều này có nghĩa là khi giá vàng tăng, chỉ số S&P 500 cũng có xu hướng tăng và ngược lại.

Giá Vàng (Close_Gold) và Giá Dầu (Close_Oil) tương quan yếu dương (0.21). Điều này có nghĩa là có rất ít mối liên hệ giữa giá vàng và giá dầu.

Giá Vàng (Close_Gold) và Chỉ Số Đô La Mỹ (Close_DollarIndex) tương quan yếu dương (0.32). Điều này cho thấy giá vàng và chỉ số Đô La Mỹ có một mối liên hệ nhẹ, khi một trong hai tăng, cái còn lại có xu hướng tăng nhẹ.

- *Kết luận:*

*Tương quan dương mạnh với S&P 500:* cho thấy giá vàng và chỉ số S&P 500 có thể cùng chịu ảnh hưởng bởi các yếu tố như lạm phát, chính sách tiền tệ, hoặc tâm lý thị trường.

*Tương quan yếu với Dầu và Đô La Mỹ*: cho thấy giá vàng và giá dầu thường không biến động cùng chiều mạnh mẽ, và giá vàng cũng không có mối liên hệ chặt chẽ với chỉ số Đô La Mỹ trong giai đoạn này.

### *3.2.4. Phân tích hồi quy*

#### *a. Giới thiệu về Mạng nơ-ron hồi quy LSTM (Long Short-Term Memory)[3]*

- **Tổng quan về mạng nơron hồi quy nhân tạo (RNN)**

Mạng nơron hồi quy (Recurrent Neural Networks - RNN) là một loại mạng nơron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu tuần tự, nơi các giá trị hiện tại bị ảnh hưởng bởi các giá trị trước đó. Đặc điểm nổi bật của RNN so với mạng nơron truyền thống là khả năng ghi nhớ và xử lý thông tin từ các bước thời gian trước đó nhờ cơ chế hồi tiếp, giúp nó trở thành công cụ mạnh mẽ cho các bài toán phân tích chuỗi thời gian, nhận dạng ngôn ngữ, hoặc xử lý giọng nói.

Tuy nhiên, RNN gặp phải hạn chế nghiêm trọng gọi là *vanishing gradien***t** khi học trên các chuỗi dữ liệu dài. Điều này dẫn đến mất mát thông tin dài hạn và giảm hiệu quả trong việc học các mối quan hệ phức tạp qua thời gian. Để giải quyết vấn đề này, mạng LSTM (Long Short-Term Memory) được ra đời, mang lại một bước tiến lớn trong việc xử lý dữ liệu chuỗi.

- **LSTM là gì?**

LSTM (Long Short-Term Memory) là một biến thể đặc biệt của RNN, được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997. Mục tiêu chính

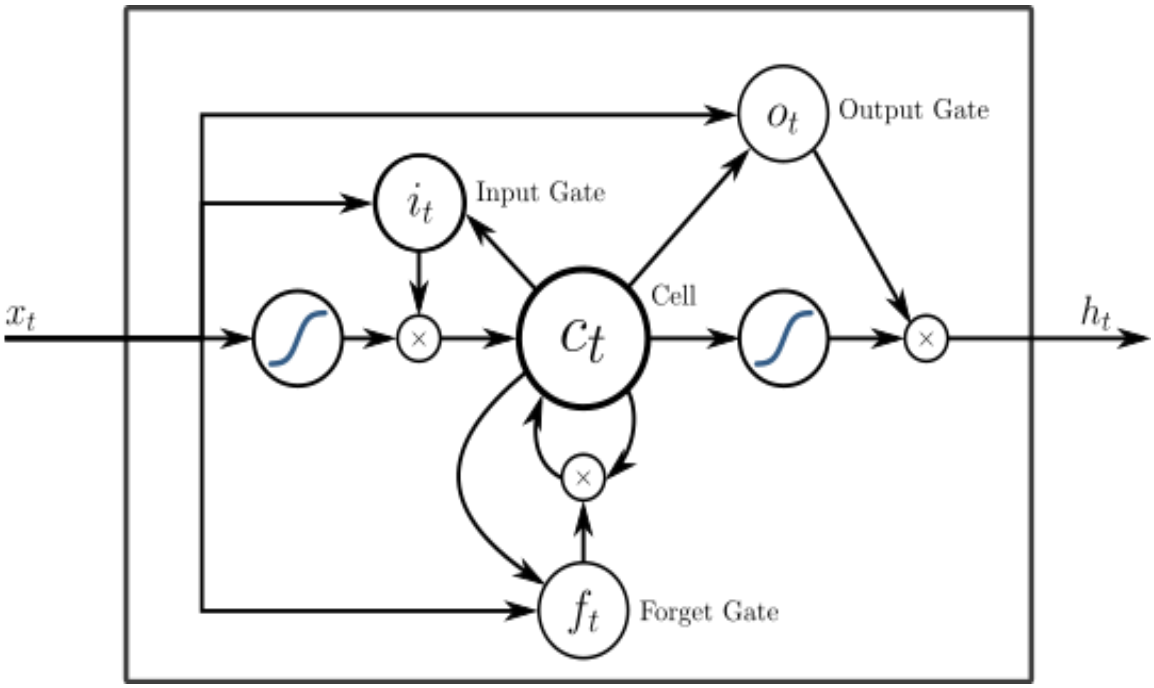
của LSTM là giải quyết các vấn đề liên quan đến vanishing gradient, cho phép mạng nơron ghi nhớ và sử dụng thông tin trong khoảng thời gian dài hơn.

Điểm cốt lõi trong thiết kế của LSTM là cấu trúc *ô nhớ (cell)* và các *bộ cổng (gates)* thông minh. Cơ chế này giúp LSTM quyết định thông tin nào cần giữ lại, loại bỏ, hoặc sử dụng để tính toán, từ đó tăng khả năng học và dự đoán các mối quan hệ phức tạp trong dữ liệu.

- **Cấu trúc của LSTM**

  LSTM có ba thành phần chính:

  - *Ô nhớ (Cell State):* Đây là thành phần chính để lưu trữ thông tin dài hạn. Có thể hình dung nó như một "đường dẫn" mà thông tin có thể di chuyển qua mà ít bị thay đổi.
  - *Bộ cổng (Gates):* LSTM sử dụng ba loại cổng để kiểm soát luồng thông tin:
    + *Cổng quên (Forget Gate):* Quyết định thông tin nào từ ô nhớ trước đó cần loại bỏ.
    + *Cổng đầu vào (Input Gate):* Quyết định thông tin mới nào nên được thêm vào ô nhớ.
    + *Cổng đầu ra (Output Gate):* Quyết định thông tin nào từ ô nhớ sẽ được đưa ra làm đầu ra.
  - *Trạng thái ẩn (Hidden State):* Đại diện cho thông tin ngắn hạn được sử dụng tại thời điểm hiện tại.




*Hình 3.15. Cấu trúc của LSTM [4]*

- **Cách hoạt động của LSTM**

Hoạt động của LSTM diễn ra thông qua các bước sau, trong đó mỗi cổng đóng vai trò riêng biệt để xử lý thông tin.

- *Các ký hiệu cần nắm rõ:*

$x_t$: Đầu vào tại thời điểm t.

$h_{t-1}$: Trạng thái ẩn từ thời điểm trước đó.

$C_{t-1}$: Trạng thái ô nhớ tại thời điểm trước đó.

$C_t$: Trạng thái ô nhớ hiện tại.

$W_f, W_i, W_c, W_o$ : Ma trận trọng số áp dụng cho các cổng.

$b_f, b_i, b_c, b_o$: Giá trị bias (độ chệch).

$\sigma$: Hàm sigmoid, đưa giá trị về khoảng [0, 1].

$tanh$: Hàm tanh, đưa giá trị về khoảng [-1, 1].

$f_t, i_t, o_t$: Đầu ra của các cổng Forget, Input và Output.

- *Bước 1: Cổng quên (Forget Gate)*

Cổng quên xác định thông tin nào từ ô nhớ trước đó $C_{t-1}$ cần được loại bỏ. Công thức tính như sau:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$f_t$: Tỉ lệ thông tin giữ lại (gần 0 là loại bỏ, gần 1 là giữ lại).

$W_t$: Ma trận trọng số áp dụng lên trạng thái ẩn trước đó và đầu vào.

$b_f$: Bias để hiệu chỉnh cổng.

- *Bước 2: Cổng đầu vào (Input Gate)*

Cổng đầu vào xác định thông tin mới cần thêm vào ô nhớ. Công thức gồm hai phần:

Quyết định phần trăm thông tin mới:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Tạo trạng thái ứng viên:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$i_t$: Mức độ chấp nhận thông tin mới.

$\tilde{C}_t$: Ứng viên cho trạng thái ô nhớ mới.

- *Bước 3: Cập nhật ô nhớ (Cell State)*

Trạng thái ô nhớ $C_t$ được cập nhật bằng cách kết hợp thông tin cần giữ lại và thông tin mới:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

$\tilde{C}_t$: Trạng thái ô nhớ mới.

- *Bước 4: Cổng đầu ra (Output Gate)*

Cuối cùng, cổng đầu ra xác định thông tin nào từ ô nhớ sẽ được sử dụng làm đầu ra tại thời điểm hiện tại:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

$o_t$: Tỉ lệ thông tin được lấy từ ô nhớ.

$h_t$: Trạng thái ẩn mới, được sử dụng làm đầu ra hoặc đầu vào cho bước tiếp theo.

- **Ứng dụng của LSTM trong phân tích dữ liệu chuỗi thời gian**

LSTM đã chứng minh sức mạnh vượt trội trong các bài toán liên quan đến dữ liệu chuỗi thời gian, như:

*Dự báo chuỗi thời gian:* Dự đoán giá vàng, giá cổ phiếu, hoặc lưu lượng giao thông.

*Phân tích tài chính:* Dự đoán rủi ro, phân tích biến động thị trường dựa trên dữ liệu lịch sử.

*Phân tích cảm xúc và ngôn ngữ*: LSTM có thể học và hiểu các phụ thuộc trong chuỗi dữ liệu văn bản.

*Phát hiện bất thường:* Phân tích tín hiệu cảm biến hoặc dữ liệu máy móc để phát hiện các lỗi tiềm ẩn.

*Ứng dụng trong y tế:* Phân tích dữ liệu y tế như điện tâm đồ (ECG) hoặc dự đoán tình trạng bệnh nhân.

***b. Huấn luyện mô hình***

- **Chuẩn bị dữ liệu**

```
[ ]  df = pd.read_csv("/content/sample_data/Data_gold_oil_dollar_sp500.csv")
```

```
[ ]  df["Date"] = pd.to_datetime(df["Date"])  # Chuyển đổi Date sang kiểu datetime
     df.set_index("Date", inplace=True)
```

Dữ liệu được thu thập từ tệp “Data_gold_oil_dollar_sp500.csv” bao gồm các chỉ số SP500, giá dầu, chỉ số đồng USD giá vàng theo ngày,...

```
# Lọc các cột cần thiết
data_filtered = df[['SP500', 'Close_Oil', 'DollarIndex', 'Close_Gold']].dropna()

# Chuẩn hóa dữ liệu
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data_filtered)

# Chia thành tập train/test (80% train, 20% test)
train_size = int(len(data_scaled) * 0.8)
train_data = data_scaled[:train_size]
test_data = data_scaled[train_size:]

# Tách input (X) và output (y)
def create_sequences(data, time_steps=1):
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:i + time_steps])  # Các cột đầu vào
        y.append(data[i + time_steps, -1])    # Cột đầu ra (Close_Gold)
    return np.array(X), np.array(y)
```

Dữ liệu được lọc để chỉ giữ lại các đặc trưng quan trọng: chỉ số SP500, giá dầu (Close_Oil), chỉ số đồng USD (DollarIndex), và giá vàng (Close_Gold).

Sau đó, dữ liệu được chuẩn hóa về khoảng [0, 1] bằng kỹ thuật MinMaxScaler, đảm bảo tính đồng nhất về thang giá trị giữa các đặc trưng đầu vào, từ đó cải thiện độ chính xác của mô hình hồi quy.

Chia dữ liệu thành tập huấn luyện và kiểm tra, dữ liệu được chia thành hai tập: 80% dành cho huấn luyện và 20% dành cho kiểm tra. Tập huấn luyện được sử dụng để dạy mô hình, trong khi tập kiểm tra được dùng để đánh giá hiệu quả dự đoán.

Tiếp theo đó, dữ liệu được chuyển đổi thành các chuỗi thời gian với độ dài 10 bước, trong đó mỗi chuỗi bao gồm các giá trị của các biến đầu vào trong 10 ngày trước đó. Đầu ra được chọn là giá vàng của ngày kế tiếp. Việc xây dựng chuỗi thời gian giúp mô hình học được mối quan hệ giữa các biến đầu vào và giá vàng trong một khoảng thời gian gần.

**Kết quả xử lý dữ liệu:**

```
# Kích thước tập dữ liệu sau xử lý
X_train.shape, y_train.shape, X_test.shape, y_test.shape
print("Dữ liệu đầu vào (X_train):", X_train[:2])  # In 2 mẫu đầu tiên của tập train
print("Dữ liệu đầu ra (y_train):", y_train[:2])  # In 2 nhãn đầu tiên của tập train
print("Shape của X_train:", X_train.shape)       # In kích thước của X_train
print("Shape của y_train:", y_train.shape)
```

Sau khi xử lý, dữ liệu huấn luyện gồm 10 bước thời gian và 4 đặc trưng đầu vào, tương ứng với các biến SP500, giá dầu, chỉ số đồng USD, và giá vàng. Dữ liệu đầu ra là giá vàng dự đoán tại thời điểm kế tiếp.

**Kết quả chạy chương trình:**

```
Dữ liệu đầu vào (X_train): [[[0.02280214 0.73848633 0.11529067 0.03912281]
  [0.02353116 0.74009794 0.11577919 0.03935673]
  [0.02365921 0.74883779 0.11089399 0.04976608]
  [0.02459888 0.74561458 0.12212995 0.04812865]
  [0.02527834 0.74617244 0.11089399 0.05111111]
  [0.02569139 0.74474679 0.09892526 0.05842105]
  [0.0234692  0.73402343 0.09794822 0.04567251]
  [0.02542291 0.72695717 0.09477284 0.05005848]
  [0.02599704 0.72534556 0.09233024 0.05368421]
  [0.02342997 0.71672968 0.10674157 0.04637427]]

 [[0.02353116 0.74009794 0.11577919 0.03935673]
  [0.02365921 0.74883779 0.11089399 0.04976608]
  [0.02459888 0.74561458 0.12212995 0.04812865]
  [0.02527834 0.74617244 0.11089399 0.05111111]
  [0.02569139 0.74474679 0.09892526 0.05842105]
  [0.0234692  0.73402343 0.09794822 0.04567251]
  [0.02542291 0.72695717 0.09477284 0.05005848]
  [0.02599704 0.72534556 0.09233024 0.05368421]
  [0.02342997 0.71672968 0.10674157 0.04637427]
  [0.02636258 0.72305213 0.11089399 0.0519883 ]]]
Dữ liệu đầu ra (y_train): [0.0519883  0.03596491]
Shape của X_train: (2971, 10, 4)
Shape của y_train: (2971,)
```

- **Xây dựng mô hình**

```
# Xây dựng mô hình LSTM
model = Sequential([
    # Lớp LSTM đầu tiên với 50 nơ-ron, trả về chuỗi (return_sequences=True)
    LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    Dropout(0.2),  # Thêm Dropout để giảm overfitting

    # Lớp LSTM thứ hai với 50 nơ-ron, không trả về chuỗi (return_sequences=False)
    LSTM(50, return_sequences=False),
    Dropout(0.2),  # Thêm Dropout để giảm overfitting

    # Lớp Dense với 25 nơ-ron, chuyển đổi đầu ra từ LSTM thành dạng đơn giản hơn
    Dense(25),

    # Lớp Dense đầu ra với 1 nơ-ron, dự đoán giá Close_Gold
    Dense(1)
])
```

Nhóm chúng em sử dụng LSTM (Long Short-Term Memory) để dự đoán giá vàng dựa trên dữ liệu đã được chuẩn bị trước đó để đảm bảo các giá trị đầu vào không bị lệch chuẩn. Mô hình sử dụng hai lớp LSTM với dropout để giảm thiểu hiện tượng overfitting và một lớp Dense để chuyển đổi đầu ra cuối cùng.

Sau khi huấn luyện, mô hình được đánh giá bằng các chỉ số độ chính xác như MAE và RMSE, giúp đo lường sự chính xác của dự đoán.

+ Lớp đầu tiên (LSTM(50, return_sequences=True)): Lớp LSTM đầu tiên có 50 nơ-ron. Tham số "return_sequences=True" giúp lớp này trả về toàn bộ chuỗi để tiếp tục xử lý bởi lớp LSTM tiếp theo.
+ Lớp Dropout (Dropout(0.2)): Dropout được thêm vào để giảm thiểu hiện tượng overfitting (quá khớp). Nó tạm thời loại bỏ 20% nơ-ron trong mỗi bước huấn luyện.
+ Lớp LSTM thứ hai (LSTM(50, return_sequences=False)): Lớp LSTM thứ hai tiếp tục học các đặc trưng từ chuỗi đầu vào nhưng không trả về toàn bộ chuỗi mà chỉ giữ lại đầu ra cuối cùng.
+ Lớp Dense (Dense(25)**):** Lớp này giảm số chiều của dữ liệu đầu ra từ lớp LSTM, giúp mô hình đơn giản hơn.
+ Lớp Dense cuối cùng (Dense(1)): Lớp cuối cùng chỉ chứa 1 nơ-ron, dự đoán giá vàng (Close_Gold).

- **Biên dịch và huấn luyện mô hình**

```
# Biên dịch mô hình
# loss='mean_squared_error': Sử dụng hàm MSE để tính sai số
# optimizer='adam': Sử dụng thuật toán Adam để tối ưu
model.compile(optimizer='adam', loss='mean_squared_error')

# Huấn luyện mô hình với dữ liệu train
# validation_data=(X_test, y_test): Sử dụng tập test để đánh giá trong quá trình huấn luyện
# epochs=20: Số lần lặp huấn luyện toàn bộ dữ liệu
# batch_size=32: Số mẫu xử lý cùng lúc trong mỗi bước lặp
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=20, batch_size=32, verbose=1)
```

- Biên dịch mô hình:

+ Hàm mất mát (loss='mean_squared_error'): Sử dụng MSE (Mean Squared Error) để đo độ chênh lệch giữa giá trị thực tế và dự đoán.
+ Bộ tối ưu hóa (optimizer='adam'): Sử dụng thuật toán Adam để điều chỉnh trọng số mạng nhằm tối ưu hóa mô hình.

- Huấn luyện mô hình (model.fit**):**

+ "epochs=20": Mô hình học qua 20 lần toàn bộ dữ liệu.
+ "batch_size=32": Dữ liệu được chia thành các lô nhỏ gồm 32 mẫu trong mỗi bước huấn luyện.

+ “validation_data=(X_test, y_test)”: Sử dụng tập kiểm thử để đánh giá mô hình trong quá trình huấn luyện.

- **Tính toán độ chính xác**

```
# Dự đoán với tập test
y_pred = model.predict(X_test)

# Chuyển đổi giá trị dự đoán và thực tế về dạng gốc (không chuẩn hóa)
y_test_original = scaler.inverse_transform(
    np.hstack((np.zeros((y_test.shape[0], 3)), y_test.reshape(-1, 1))))[:, -1]
y_pred_original = scaler.inverse_transform(
    np.hstack((np.zeros((y_pred.shape[0], 3)), y_pred)))[:, -1]

# Tính toán sai số
mae = mean_absolute_error(y_test_original, y_pred_original)
rmse = np.sqrt(mean_squared_error(y_test_original, y_pred_original))

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
```

MAE (Mean Absolute Error): Tính sai số tuyệt đối trung bình giữa giá trị thực tế và dự đoán.

RMSE (Root Mean Squared Error): Đo sai số bình phương trung bình, là thước đo phổ biến trong dự đoán chuỗi thời gian.

- **Kết quả:**

Epoch 1/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **5s** 15ms/step - loss: 0.0141 - val_loss: 8.1672e-04

Epoch 2/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 0.0012 - val_loss: 0.0017

Epoch 3/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 0.0010 - val_loss: 0.0011

Epoch 4/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 15ms/step - loss: 7.8119e-04 - val_loss: 0.0013

Epoch 5/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 19ms/step - loss: 6.7775e-04 - val_loss: 0.0040

Epoch 6/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **3s** 24ms/step - loss: 6.4790e-04 - val_loss: 0.0021

Epoch 7/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.9942e-04 - val_loss: 0.0025

Epoch 8/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.8145e-04 - val_loss: 0.0041

Epoch 9/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.6032e-04 - val_loss: 0.0022

Epoch 10/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.2208e-04 - val_loss: 0.0026

Epoch 11/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 4.8353e-04 - val_loss: 0.0032

Epoch 12/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.0758e-04 - val_loss: 0.0022

Epoch 13/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 5.1660e-04 - val_loss: 0.0011

Epoch 14/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 12ms/step - loss: 4.8870e-04 - val_loss: 0.0035

Epoch 15/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 19ms/step - loss: 4.4132e-04 - val_loss: 0.0029

Epoch 16/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 18ms/step - loss: 4.5317e-04 - val_loss: 5.0347e-04

Epoch 17/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 20ms/step - loss: 4.6594e-04 - val_loss: 0.0018

Epoch 18/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 12ms/step - loss: 4.5586e-04 - val_loss: 8.2945e-04

Epoch 19/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **2s** 16ms/step - loss: 4.3910e-04 - val_loss: 0.0029

Epoch 20/20

**93/93** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **3s** 29ms/step - loss: 4.2635e-04 - val_loss: 7.4016e-04

**23/23** ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 11ms/step

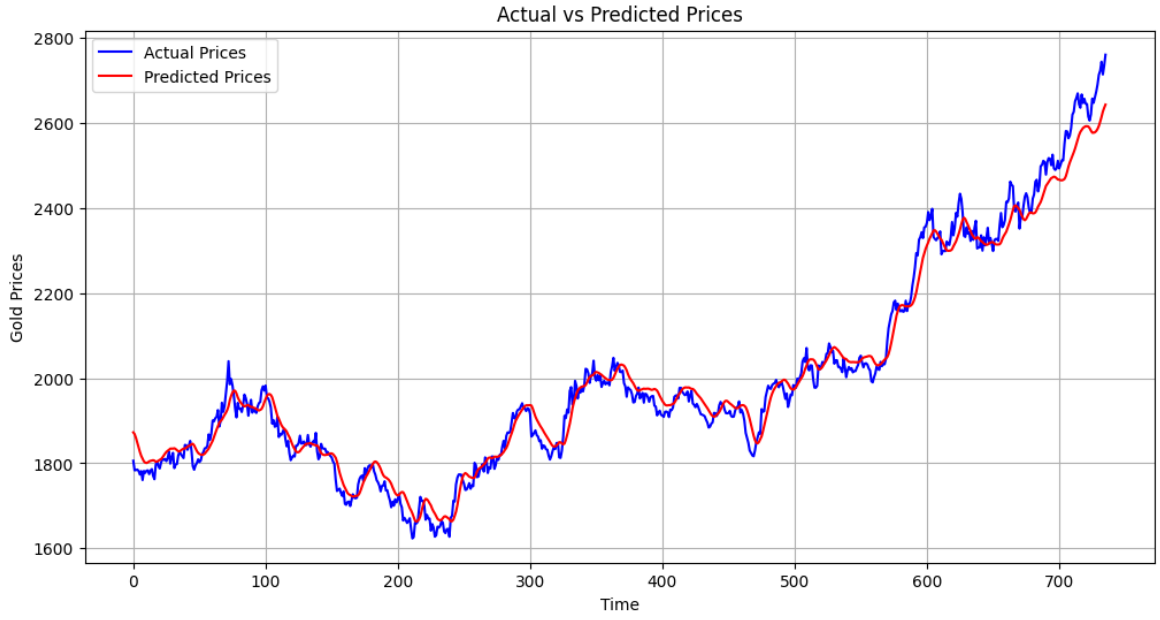
Mean Absolute Error (MAE): 34.37662184803703

Root Mean Squared Error (RMSE): 46.522090716385726

- **Trực quan hóa kết quả:**

```
# Vẽ biểu đồ so sánh Actual và Predicted
def plot_actual_vs_predicted(y_actual, y_predicted):
    """
    Vẽ biểu đồ so sánh giá thực tế và giá dự đoán.
    y_actual: Giá trị thực tế (actual values).
    y_predicted: Giá trị dự đoán (predicted values).
    """
    plt.figure(figsize=(12, 6))
    plt.plot(y_actual, label="Actual Prices", color="blue")
    plt.plot(y_predicted, label="Predicted Prices", color="red")
    plt.title("Actual vs Predicted Prices")
    plt.xlabel("Time")
    plt.ylabel("Gold Prices")
    plt.legend()
    plt.grid()
    plt.show()

# Gọi hàm vẽ biểu đồ với tập test
plot_actual_vs_predicted(y_test_original, y_pred_original)
```

Để trực quan hóa mô hình, chúng em sử dụng biểu đồ đường để so sánh dữ liệu thực tế và dữ liệu dự đoán.




*Hình 3.16. Biểu đồ so sánh giá vàng thực tế và giá vàng dự báo*

Biểu đồ này hiển thị một cách trực quan sự khác biệt, hoặc sự tương đồng, giữa hai tập giá trị, giúp người dùng đánh giá hiệu suất của mô hình một cách dễ dàng. Đường màu xanh biểu diễn giá trị thực tế, trong khi đường màu đỏ biểu diễn các giá trị dự đoán, tạo nên một biểu đồ đối chiếu trực tiếp.

### *c. Thực thi mô hình*

- **Hàm dự báo linh hoạt**

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.ticker import MaxNLocator
# Hàm dự báo linh hoạt
def forecast_next_days(model, last_data, days=10, time_steps=10):
    """
    Dự báo giá trị cho một số ngày nhất định.
    model: Mô hình LSTM đã huấn luyện.
    last_data: Dữ liệu đầu vào (chuỗi thời gian gần nhất, định dạng chuẩn hóa).
    days: Số ngày muốn dự báo (mặc định 10).
    time_steps: Số bước thời gian đầu vào cho mô hình.
    """
    predictions = []  # Danh sách chứa giá trị dự đoán
    current_input = last_data[-time_steps:, :]  # Lấy chuỗi thời gian gần nhất làm đầu vào

    for _ in range(days):
        # Dự đoán giá trị tiếp theo
        predicted = model.predict(current_input[np.newaxis, :, :])[0][0]  # Dự đoán 1 giá trị
        predictions.append(predicted)  # Thêm giá trị dự đoán vào danh sách

        # Cập nhật chuỗi thời gian với giá trị dự đoán
        current_input = np.vstack((current_input[1:], [[predicted, 0, 0, 0]]))  # Đẩy dữ liệu mới vào chuỗi

    # Chuyển đổi giá trị dự đoán về giá trị gốc
    predictions_original = scaler.inverse_transform(
        np.hstack((np.zeros((len(predictions), 3)), np.array(predictions).reshape(-1, 1)))
    )[:, -1]

    return predictions_original
```

Hàm "forecast_next_days" được thiết kế để dự đoán giá vàng cho một số ngày nhất định dựa trên mô hình LSTM đã được huấn luyện trước đó. Dữ liệu đầu vào của hàm bao gồm chuỗi thời gian gần nhất, được chuẩn hóa để phù hợp với mô hình, và số ngày cần dự báo. Cách hoạt động của hàm này dựa trên một quy trình lặp, trong đó mô hình LSTM lần lượt dự đoán giá trị tiếp theo dựa trên chuỗi thời gian hiện tại. Giá trị dự đoán sau mỗi lần lặp sẽ được thêm vào cuối chuỗi và đồng thời cập nhật chuỗi thời gian đầu vào cho lần dự đoán tiếp theo. Điều này đảm bảo quá trình dự báo được thực hiện liên tục, dựa trên cả dữ liệu quá khứ và các giá trị mới nhất được mô hình dự đoán. Sau khi hoàn thành, các giá trị dự đoán được chuyển đổi từ dạng chuẩn hóa về giá trị gốc để dễ dàng so sánh và sử dụng.

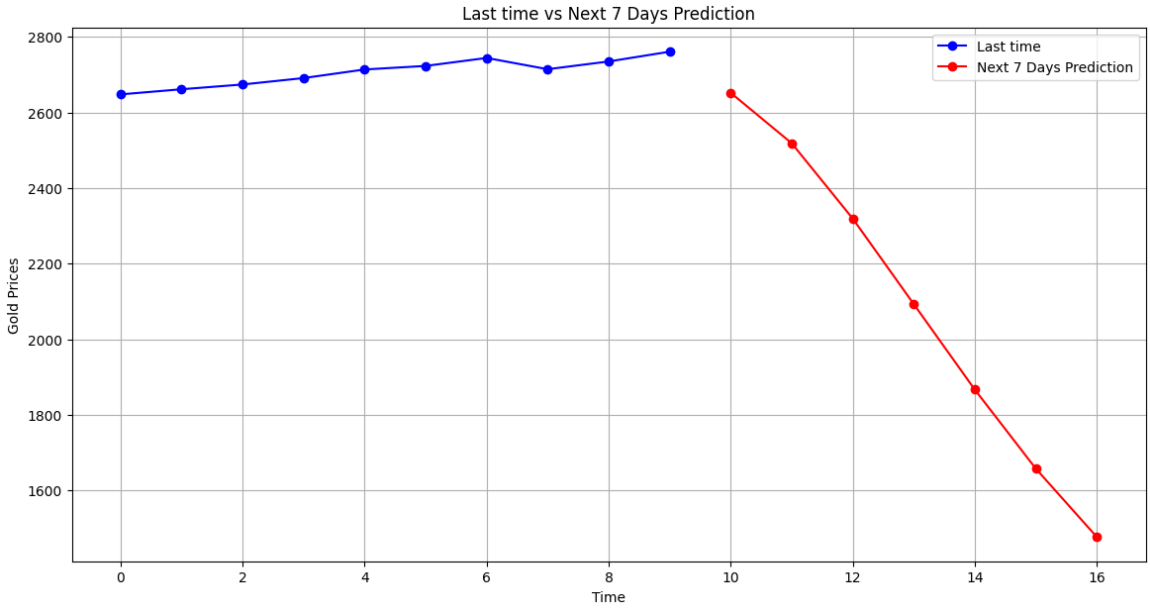
```
# Dự báo cho 7 ngày (hoặc nhiều hơn)
time_steps = 10  # Bước thời gian đầu vào
last_data = test_data[-time_steps:]  # Lấy dữ liệu cuối của tập test
next_7_days = forecast_next_days(model, last_data, days=7, time_steps=time_steps)
```

Dự báo cho 7 ngày, chúng em sử dụng bước thời gian "time_steps = 10" và lấy dữ liệu cuối của tập test để đưa vào hàm forecast_next_days dữ đoán dữ liệu cho 7 ngày.

- **Biểu đồ thể hiện kết quả dự báo**

```
# Hàm vẽ biểu đồ
def plot_predict(current_input, next_7_days):
    """
    Vẽ biểu đồ so sánh dữ liệu hiện tại và dữ liệu dự đoán.
    current_input: Dữ liệu hiện tại (input cuối cùng được sử dụng cho dự đoán).
    next_7_days: Dữ liệu dự đoán trong 7 ngày tới.
    """
    plt.figure(figsize=(14, 7))

    # Vẽ dữ liệu hiện tại (Current input)
    plt.plot(range(len(current_input)), current_input, label="Last time", color="blue", marker='o')

    # Vẽ dữ liệu dự đoán (Next 7 days)
    plt.plot(range(len(current_input), len(current_input) + len(next_7_days)),
             next_7_days, label="Next 7 Days Prediction", color="red", marker='o')
    plt.gca().xaxis.set_major_locator(MaxNLocator(integer=True))
    # Thêm thông tin biểu đồ
    plt.title("Last time vs Next 7 Days Prediction")
    plt.xlabel("Time")
    plt.ylabel("Gold Prices")
    plt.legend()
    plt.grid()
    plt.show()

# Chuyển đổi last_data về giá trị gốc để vẽ biểu đồ
current_input = scaler.inverse_transform(
    np.hstack((np.zeros((last_data.shape[0], 3)), last_data[:, -1].reshape(-1, 1))))[:, -1]
# Vẽ biểu đồ
plot_predict(current_input, next_7_days)
```

Hàm "plot_predict" dùng để vẽ biểu đồ thể hiện kết quả dự báo, tập trung vào việc minh họa dữ liệu hiện tại và xu hướng dự báo trong tương lai. Biểu đồ thể hiện sự chuyển tiếp giữa chuỗi dữ liệu thực tế (hiện tại) và chuỗi dữ liệu dự đoán cho 7 ngày tiếp theo. Phần dữ liệu thực tế, được vẽ bằng đường màu xanh, biểu thị những gì đã xảy ra, trong khi đường màu đỏ biểu diễn xu hướng dự đoán, mang lại cái nhìn trực quan về sự biến động giá vàng trong tương lai.




*Hình 3.17. Biểu đồ dự đoán giá vàng cho 7 ngày tiếp theo*

### *d. Đánh giá mô hình*

Để đánh giá mô hình, chúng em sử dụng phương pháp 10-fold cross-validation.

Phương pháp 10-fold cross-validation là một kỹ thuật phổ biến trong học máy để đánh giá hiệu quả của mô hình. Mục đích chính của phương pháp này là giảm thiểu nguy cơ overfitting (mô hình quá khớp với dữ liệu huấn luyện) và đảm bảo rằng mô hình có khả năng tổng quát hóa tốt khi áp dụng vào dữ liệu mới.

- **Quy trình hoạt động của 10-fold cross-validation:**

*Bước 1:* Chia dữ liệu thành 10 phần (folds), toàn bộ tập dữ liệu được chia ngẫu nhiên thành 10 phần bằng nhau (nếu dữ liệu không chia đều, các phần có thể khác nhau một chút về số lượng mẫu).

*Bước 2:* Lặp lại 10 lần (10 folds), trong mỗi lần lặp, một phần sẽ được giữ lại làm tập kiểm thử (validation set), trong khi 9 phần còn lại được sử dụng để huấn luyện mô hình.

*Bước 3:* Đánh giá trên tập kiểm thử, sau khi mô hình được huấn luyện trên tập huấn luyện, nó sẽ được kiểm tra trên tập kiểm thử để tính toán các chỉ số đánh giá như độ chính xác, sai số, hoặc hệ số xác định ($R^2$).

Lặp lại quá trình cho tất cả các folds: Quá trình này được thực hiện 10 lần, mỗi phần dữ liệu sẽ lần lượt được sử dụng làm tập kiểm thử một lần.

*Bước 4:* Tính toán kết quả trung bình, sau khi hoàn thành tất cả 10 lần huấn luyện và kiểm thử, các chỉ số đánh giá từ từng fold sẽ được tổng hợp và tính trung bình. Kết quả trung bình này cung cấp một ước lượng khách quan hơn về hiệu quả của mô hình.

Phương pháp 10-fold cross-validation giúp đảm bảo toàn bộ dữ liệu đều được sử dụng cả trong huấn luyện lẫn kiểm thử, giúp khai thác tối đa thông tin từ tập dữ liệu. Đưa ra kết quả đánh giá ổn định hơn so với việc chia tập dữ liệu ngẫu nhiên một lần (train/test split). Cuối cùng là giúp phát hiện và giảm thiểu nguy cơ mô hình quá khớp hoặc đánh giá sai hiệu năng trên tập kiểm thử cố định.

- **Ứng dụng với mô hình LSTM:**

```
from sklearn.metrics import mean_squared_error, r2_score
from tensorflow.keras.layers import Input
import pandas as pd
import numpy as np

# Hàm thực hiện 10-fold cross-validation và tạo bảng thống kê
def cross_validate_and_report(data, time_steps, n_splits=10, epochs=20, batch_size=32):
    from sklearn.model_selection import KFold

    # Khởi tạo KFold
    kfold = KFold(n_splits=n_splits, shuffle=True, random_state=42)

    # Lưu trữ kết quả của các fold
    fold_results = []
```

Chương trình sử dụng một số thư viện bao gồm:

+ "sklearn.metrics" cung cấp các công cụ tính toán chỉ số đánh giá như MSE (Mean Squared Error) và R2 Score.
+ "tensorflow.keras.layers" giúp xây dựng mạng nơ-ron LSTM, cho phép mô hình học các mối quan hệ theo chuỗi thời gian.

Hàm "cross_validate_and_report" thực hiện toàn bộ quy trình 10-fold cross-validation. Hàm nhận các tham số đầu vào như dữ liệu đã chuẩn hóa (data), số bước thời gian (time_steps), và số lần chia nhỏ (n_splits). Số epochs và batch size được đặt mặc định để kiểm soát quá trình huấn luyện.

Dữ liệu được chia thành 10 phần bằng cách sử dụng công cụ “Kfold” từ thư viện “sklearn”.

```
# Duyệt qua từng fold
for fold, (train_idx, val_idx) in enumerate(kfold.split(data), 1):
    # Tạo dữ liệu train và validation
    train_data = data[train_idx]
    val_data = data[val_idx]

    # Tạo chuỗi thời gian (sequences)
    X_train, y_train = create_sequences(train_data, time_steps)
    X_val, y_val = create_sequences(val_data, time_steps)

    # Xây dựng mô hình
    model = Sequential([
        Input(shape=(X_train.shape[1], X_train.shape[2])),
        LSTM(50, return_sequences=True),
        Dropout(0.2),
        LSTM(50, return_sequences=False),
        Dropout(0.2),
        Dense(25),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mean_squared_error')

    # Huấn luyện mô hình
    model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size, verbose=0)

    # Dự đoán trên tập validation
    y_val_pred = model.predict(X_val)

    # Chuyển giá trị dự đoán và thực tế về dạng gốc
    y_val_original = scaler.inverse_transform(
        np.hstack((np.zeros((y_val.shape[0], 3)), y_val.reshape(-1, 1)))
    )[:, -1]
    y_val_pred_original = scaler.inverse_transform(
        np.hstack((np.zeros((y_val_pred.shape[0], 3)), y_val_pred))
    )[:, -1]

    # Tính các chỉ số
    mse = mean_squared_error(y_val_original, y_val_pred_original)
    mae = mean_absolute_error(y_val_original, y_val_pred_original)
    r2 = r2_score(y_val_original, y_val_pred_original)

    # Lưu kết quả của fold
    fold_results.append({'Fold': fold, 'Mean Squared Error': mse, 'R2 Score': r2,
                         "Mean Absolute Error" : mae})
```

Trong mỗi fold, một phần dữ liệu được sử dụng để kiểm thử, còn lại dùng để huấn luyện. Hàm “create_sequences” tạo ra các chuỗi thời gian từ dữ liệu gốc, giúp mô hình LSTM học được các mẫu xu hướng trong dữ liệu.

Mô hình LSTM là phần chính của chương trình, được thiết kế để xử lý dữ liệu chuỗi thời gian và dự đoán giá trị tương lai.

+ Lớp “LSTM” với 50 đơn vị và tham số “return_sequences=True” giúp trích xuất các đặc trưng liên quan đến chuỗi thời gian.
+ Lớp “Dropout” với tỷ lệ 0.2 giúp giảm nguy cơ overfitting bằng cách bỏ qua một số nơ-ron trong quá trình huấn luyện.
+ Lớp “Dense” cuối cùng giảm đầu ra thành một giá trị duy nhất, tương ứng với dự đoán giá vàng.

Mô hình được huấn luyện trên tập huấn luyện qua một số epochs cố định. Sau khi huấn luyện, mô hình được sử dụng để dự đoán trên tập kiểm thử.

Giá trị dự đoán và giá trị thực tế được chuyển đổi về dạng gốc để so sánh nhờ hàm “scaler.inverse_transform.”

Các chỉ số MSE (Mean Squared Error), MAE (Mean Absolute Error), và R2 Score được sử dụng để đo lường hiệu quả của mô hình. Kết quả của từng fold được lưu trữ, sau đó tính toán trung bình để đánh giá tổng thể hiệu năng của mô hình.

```
    # Tạo DataFrame từ kết quả
    results_df = pd.DataFrame(fold_results)

    # Thêm dòng trung bình
    avg_row = pd.DataFrame({
    'Fold': ['Average'],
    'Mean Squared Error': [results_df['Mean Squared Error'].mean()],
    'R2 Score': [results_df['R2 Score'].mean()],
    "Mean Absolute Error" : [results_df["Mean Absolute Error"].mean()]
    })

    # Nối dòng trung bình với DataFrame kết quả
    results_df = pd.concat([results_df, avg_row], ignore_index=True)

    # In bảng kết quả
    print(results_df)

    return results_df
```

Bảng kết quả hiển thị chỉ số của từng fold và trung bình, giúp nhận định rõ hơn về hiệu quả của mô hình.

Triển khai chương trình, hàm được gọi với tham số cụ thể như sau:

```
# Thực hiện cross-validation và tạo bảng thống kê
results_df = cross_validate_and_report(data_scaled, time_steps=10,
                                       n_splits=10, epochs=10, batch_size=32)
```

- **Kết quả:**

Kết quả đánh giá mô hình:

```
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 38ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 26ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 41ms/step
```

| | Fold | Mean Squared Error | R2 Score | Mean Absolute Error |
|---|---|---|---|---|
| 0 | 1 | 7704.892696 | 0.929729 | 67.700423 |
| 1 | 2 | 9187.002131 | 0.911453 | 68.967285 |
| 2 | 3 | 7318.078468 | 0.932017 | 65.080494 |
| 3 | 4 | 10452.081513 | 0.909702 | 77.512461 |
| 4 | 5 | 7184.565853 | 0.938248 | 63.908963 |
| 5 | 6 | 9924.994457 | 0.901461 | 75.774456 |
| 6 | 7 | 7923.735041 | 0.934470 | 66.922799 |
| 7 | 8 | 7897.596025 | 0.907085 | 68.161743 |
| 8 | 9 | 8880.085630 | 0.919157 | 71.321746 |
| 9 | 10 | 7898.606416 | 0.929003 | 67.065293 |
| 10 | Average | 8437.163823 | 0.921233 | 69.241566 |

*Hình 3.18. Kết quả đánh giá mô hình*

Từ bảng kết quả, chúng ta có thể đánh giá tổng quan về mô hình:

- *Độ chính xác của mô hình:* chỉ số "R2 Score" trung bình đạt 0.9212, cho thấy mô hình có khả năng giải thích hơn 92% sự biến động trong dữ liệu thực. Các giá trị R2 của từng fold đều trên 0.90, phản ánh sự ổn định của mô hình trên các tập kiểm thử khác nhau.
- *Sai số của mô hình:* "MSE" trung bình là 8437.16, và "MAE" trung bình là 69.24 (chênh lệch trung bình khoảng 69 đơn vị so với giá trị thực tế), cho thấy sai số giữa giá trị dự đoán và thực tế không quá lớn.
- *Sự ổn định và tính tổng quát của mô hình:* Kết quả cho thấy sự chênh lệch giữa các fold là không quá lớn. Ví dụ:

+ Fold có MSE thấp nhất là 7318 (Fold 3), trong khi fold có MSE cao nhất là 10452 (Fold 4).
+ Fold có MAE thấp nhất là 63.91 (Fold 5), và fold có MAE cao nhất là 77.51 (Fold 4).

## 3.3. Kết luận chương 3

Trong chương 3, nhóm đã thực hiện toàn bộ quy trình thực nghiệm, bao gồm tiền xử lý dữ liệu, phân tích mô tả và triển khai mô hình LSTM để dự báo giá vàng. Dữ liệu được làm sạch, chuẩn hóa và chuyển đổi thành các chuỗi thời gian để phù hợp với yêu cầu của mô hình. Các phân tích mô tả đã làm rõ mối quan hệ giữa giá vàng và các yếu tố kinh tế như chỉ số USD, giá dầu, và chỉ số S&P 500, cung cấp cái nhìn sâu sắc về xu hướng biến động của giá vàng.

Mô hình LSTM đã được triển khai hiệu quả, với khả năng dự báo giá vàng tương đối chính xác. Điều này được minh chứng qua các chỉ số đánh giá như MAE, RMSE và biểu đồ so sánh giá vàng thực tế với giá vàng dự báo. Kết quả này cho thấy mô hình có khả năng nắm bắt tốt các xu hướng chính của dữ liệu, hỗ trợ đắc lực cho mục tiêu dự báo.

# CHƯƠNG 4. XÂY DỰNG SẢN PHẨM

## 4.1. Công cụ và công nghệ sử dụng

*Giới thiệu về Tkinter:*

Tkinter là một thư viện Python tiêu chuẩn để tạo giao diện người dùng đồ họa (GUI) một cách nhanh chóng và tiện lợi. Thay vì phải làm việc với HTML hay CSS, bạn có thể tận dụng ngôn ngữ Python để tạo ra giao diện người dùng một cách đơn giản.

Với Tkinter, bạn dễ dàng thiết lập và tùy chỉnh các thành phần giao diện như nhãn, nút bấm, khung nhập liệu, hộp thoại và nhiều tùy chọn khác. Thư viện này đảm bảo rằng việc phát triển giao diện người dùng trở nên nhanh chóng và hiệu quả.

Nhờ sự linh hoạt và khả năng hỗ trợ đa nền tảng, Tkinter trở thành lựa chọn hàng đầu cho những ai muốn nhanh chóng triển khai các ứng dụng desktop. Bạn cũng có thể tích hợp các thanh cuộn, biểu đồ, danh sách, và nhiều thành phần khác mà không cần có kiến thức chuyên sâu về giao diện.

Tkinter đang được sử dụng rộng rãi trong lĩnh vực phát triển ứng dụng desktop, nghiên cứu và giảng dạy. Có một cộng đồng người dùng phong phú cung cấp tài nguyên và hỗ trợ cho người mới, giúp bạn bắt đầu với Tkinter một cách thuận lợi.

*Giới thiệu về thư viện Joblib:*

Joblib là thư viện Python chuyên dùng cho xử lý song song và lưu trữ dữ liệu hiệu quả. Nó hữu ích trong các ứng dụng machine learning, khoa học dữ liệu và xử lý dữ liệu lớn.

- Với Joblib, chúng ta có thể:

+ Lưu trữ và tải lại object Python nhanh chóng.
+ Xử lý song song nhờ khai thác nhiều CPU core.
+ Tích hợp nén dữ liệu để tối ưu hóa dung lượng.

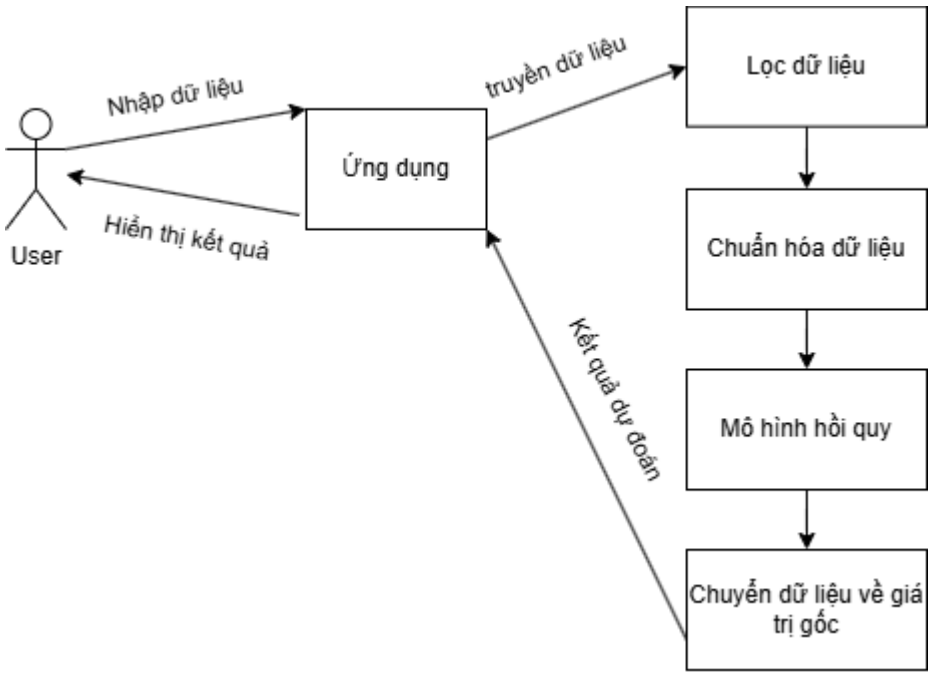
Với API thân thiện và tính tương thích cao với numpy, scipy, Joblib giúp tối ưu hóa quá trình xử lý và triển khai mô hình machine learning.

## 4.2. Chuẩn bị tài nguyên xây dựng chương trình

Khi mô hình đã được huấn luyện và dữ liệu đã được chuẩn hóa ta sẽ lưu lại chúng để phục vụ cho việc dự đoán dựa trên dữ liệu người dùng đưa vào:

```
model.save("gold_price_model.keras")
import joblib
joblib.dump(scaler, filename: "scaler.pkl")
```

## 4.3. Mô tả chương trình



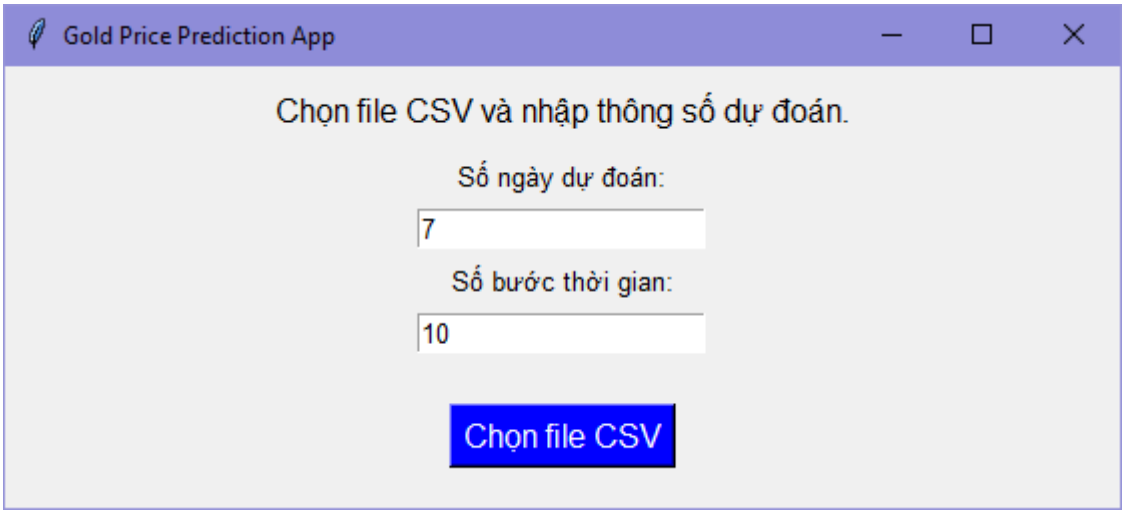

*Hình 4.1. Sơ đồ use case*

*Bảng 4.1. Bảng mô tả use case*

| | |
|---|---|
| Tên use case | Dự đoán dữ liệu với mô hình LSTM |
| Tóm tắt | Người dùng nhập dữ liệu đầu vào, hệ thống kiểm tra tính hợp lệ của dữ liệu. Nếu dữ liệu đúng, hệ thống sẽ sử dụng mô hình LSTM để dự đoán và hiển thị kết quả dưới dạng biểu đồ (chart). Nếu dữ liệu sai, thông báo lỗi sẽ được hiển thị cho người dùng. |
| Actor | Người dùng |

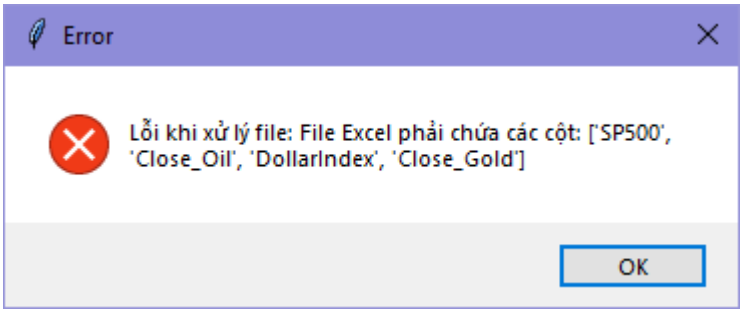
| | |
|---|---|
| Tiền điều kiện | Hệ thống đã tải mô hình dự đoán vào bộ nhớ.<br>Người dùng truy cập vào được giao diện. |
| Đảm bảo tối thiểu | Người dùng được thông báo nếu dữ liệu nhập vào không hợp lệ. |
| Đảm bảo thành công | Người dùng nhận được kết quả dự đoán dưới dạng biểu đồ nếu dữ liệu hợp lệ. |
| Kích hoạt | Người dùng mở ứng dụng và nhập dữ liệu cần dự đoán. |
| Luồng sự kiện | Luồng chính:<br>1. Người dùng mở ứng dụng.<br>2. Hệ thống tải mô hình vào bộ nhớ.<br>3. Hệ thống hiển thị giao diện cho người dùng.<br>4. Người dùng nhập dữ liệu cần dự đoán.<br>5. Hệ thống kiểm tra tính hợp lệ của dữ liệu:<br>a. Nếu dữ liệu không hợp lệ: Chuyển sang luồng thay thế (Bước 5a).<br>b. Nếu dữ liệu hợp lệ: Chuyển sang Bước 6.<br>6. Lọc dữ liệu.<br>7. Chuẩn hóa dữ liệu.<br>8. Hệ thống sử dụng mô hình để dự đoán.<br>9. Chuyển đổi dữ liệu về giá trị gốc.<br>10. Hệ thống hiển thị kết quả dự đoán dưới dạng biểu đồ.<br>Luồng thay thế **(**Bước 5a)<br>Hệ thống hiển thị thông báo lỗi cho người dùng.<br>Người dùng chỉnh sửa và nhập lại dữ liệu. |

## 4.4. Demo sản phẩm




*Hình 4.2. Giao diện chính của chương trình*

Nhập dữ liệu và chọn file csv tương ứng.




*Hình 4.3. Thông báo lỗi dữ liệu*

Lỗi khi người dùng chọn file csv không đúng format.

```
predict_label = tk.Label(root, text="Số ngày dự đoán:", font=("Arial", 10))
predict_label.pack()
predict_entry = tk.Entry(root, font=("Arial", 10))
predict_entry.pack(pady=5)
predict_entry.insert( index: 0,  string: "7")  # Giá trị mặc định

# Nhãn và ô nhập cho số bước thời gian
step_label = tk.Label(root, text="Số bước thời gian:", font=("Arial", 10))
step_label.pack()
step_entry = tk.Entry(root, font=("Arial", 10))
step_entry.pack(pady=5)
step_entry.insert( index: 0,  string: "10")  # Giá trị mặc định

# Nút chọn file
open_button = tk.Button(root, text="Chọn file CSV", command=open_file, bg="blue", fg="white", font=("Arial", 12))
open_button.pack(pady=20)
```

Nhận các giá trị tương ứng.

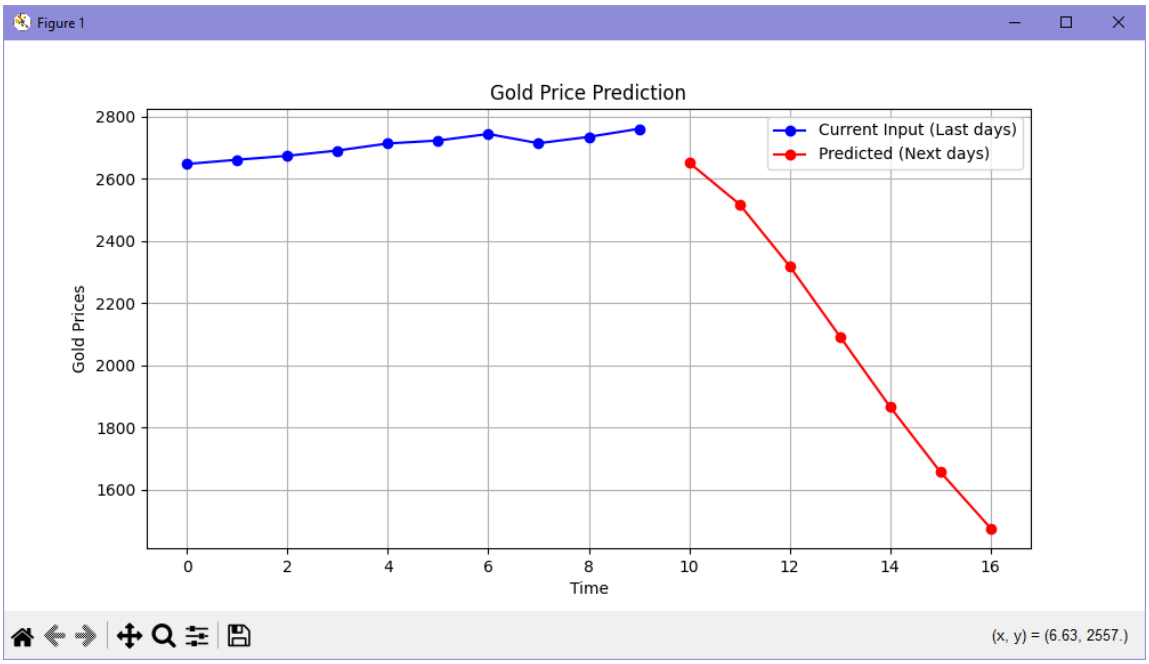
```
def open_file():
    """
    Mở hộp thoại chọn file và xử lý file được chọn.
    """
    try:
        file_path = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])
        if file_path:
            count_date_predict = int(predict_entry.get())
            count_date_step = int(step_entry.get())
            process_file(file_path, count_date_predict, count_date_step)
    except ValueError:
        messagebox.showerror( title: "Invalid Input",  message: "Hãy nhập giá trị hợp lệ cho số ngày dự đoán và số bước thời gian.")
```

Khi người dùng chọn file thành công. Ứng dụng sẽ gọi hàm process_file để xử lý file và dự đoán kết quả.

```
def process_file(file_path, count_date_predict, count_date_step):
    """
    Xử lý file Excel được kéo thả, lấy dữ liệu và dự đoán.
    """
    try:
        # Đọc file Excel
        data = pd.read_csv(file_path)
        required_columns = ['SP500', 'Close_Oil', 'DollarIndex', 'Close_Gold']

        # Kiểm tra các cột cần thiết
        if not all(col in data.columns for col in required_columns):
            raise ValueError(f"File Excel phải chứa các cột: {required_columns}")

        # Lọc dữ liệu và chuẩn hóa
        data_filtered = data[required_columns].dropna()
        data_scaled = scaler.transform(data_filtered)

        # Lấy dữ liệu cuối cùng để làm đầu vào dự đoán
        current_input = data_scaled[-count_date_step:, :]  # Bước thời gian gần nhất

        # Dự đoán count_date_predict ngày tiếp theo
        predictions = []
        for _ in range(count_date_predict):
            predicted = model.predict(current_input[np.newaxis, :, :])[0][0]
            predictions.append(predicted)
            current_input = np.vstack((current_input[1:], [[predicted, 0, 0, 0]]))

        # Chuyển giá trị dự đoán về giá trị gốc
        predictions_original = scaler.inverse_transform(
            np.hstack((np.zeros((len(predictions), 3)), np.array(predictions).reshape(-1, 1)))
        )[:, -1]

        # Vẽ biểu đồ
        plot_predictions(data_filtered['Close_Gold'][-count_date_step:].values, predictions_original)

    except Exception as e:
        messagebox.showerror( title: "Error",  message: f"Lỗi khi xử lý file: {str(e)}")
```

Khi đã hoàn tất việc dự đoán và chuyển đổi dữ liệu về giá trị gốc. Ứng dụng sẽ thực hiện vẽ đồ thị và trả về cho người dùng.

```python
def plot_predictions(current_input, predictions):
    """
    Vẽ biểu đồ dữ liệu hiện tại và dự đoán.
    """
    plt.figure(figsize=(10, 5))
    plt.plot( *args: range(len(current_input)), current_input, label="Current Input (Last days)", color="blue", marker="o")
    plt.plot(
        *args: range(len(current_input), len(current_input) + len(predictions)), predictions,
        label="Predicted (Next days)", color="red", marker="o"
    )
    plt.title("Gold Price Prediction")
    plt.xlabel("Time")
    plt.ylabel("Gold Prices")
    plt.legend()
    plt.grid()
    plt.show()
```




*Hình 4.4. Kết quả dự đoán*

Đường kẻ màu xanh tượng trưng cho khoảng thời gian lấy để dự đoán. Và những chấm xanh biểu thị cho từng ngày. Tiếp theo màu đỏ tượng trưng cho thời gian dự đoán và những chấm đỏ biểu thị cho những ngày tiếp theo được dự đoán kể từ dấu chấm cuối của đường kẻ màu xanh.

# KẾT LUẬN

Trong khuôn khổ đề tài "Phân tích mô tả thị trường vàng và dự báo giá vàng bằng mô hình hồi quy", nhóm chúng em đã tập trung nghiên cứu một trong những thị trường tài chính quan trọng nhất hiện nay – thị trường vàng. Mục tiêu chính của đề tài là phân tích các yếu tố ảnh hưởng đến giá vàng, từ đó xây dựng một mô hình dự báo có khả năng hỗ trợ nhà đầu tư đưa ra quyết định chiến lược. Chúng em đã chọn mạng nơ-ron LSTM, một công cụ mạnh mẽ trong xử lý chuỗi thời gian, để dự đoán xu hướng biến động của giá vàng dựa trên dữ liệu thực nghiệm.

Nhóm đã thực hiện đầy đủ các giai đoạn nghiên cứu, từ việc xác định mục tiêu, thu thập và tiền xử lý dữ liệu, đến phân tích mô tả và hồi quy. Chúng em đã sử dụng bộ dữ liệu bao gồm các chỉ số kinh tế quan trọng như giá dầu, chỉ số đồng USD (Dollar Index), và chỉ số S&P500 – những yếu tố được biết đến có tác động mạnh mẽ đến giá vàng. Quá trình phân tích mô tả đã giúp nhận diện các đặc điểm chính của dữ liệu, phát hiện các xu hướng tiềm năng và loại bỏ các điểm dữ liệu ngoại lệ có thể làm giảm độ chính xác của mô hình.

Trong giai đoạn hồi quy, chúng em đã triển khai mô hình mạng nơ-ron LSTM với cấu trúc linh hoạt để nắm bắt các mẫu phức tạp từ dữ liệu chuỗi thời gian. Kết quả dự báo không chỉ khớp với thực tế mà còn cung cấp cái nhìn rõ ràng về xu hướng giá vàng trong tương lai gần.

Dù đạt được nhiều thành quả, chúng em nhận thấy nghiên cứu vẫn tồn tại một số hạn chế. Dữ liệu sử dụng chủ yếu tập trung vào các chỉ số thị trường tài chính, chưa mở rộng đến các yếu tố kinh tế vĩ mô khác như lãi suất ngân hàng, lạm phát, hoặc tác động từ các sự kiện chính trị toàn cầu. Những yếu tố này có thể làm rõ hơn mối quan hệ đa chiều giữa giá vàng và các biến động kinh tế.

Ngoài ra, quá trình tối ưu hóa mô hình vẫn cần nhiều thử nghiệm hơn để cải thiện độ chính xác, đặc biệt trong việc dự đoán các biến động bất thường hoặc đột ngột. Hơn nữa, chúng em chưa có cơ hội triển khai ứng dụng mô hình vào thực tế, như xây dựng một công cụ dự báo trực tuyến, để đánh giá khả năng hoạt động của mô hình trong môi trường thực.

Trong tương lai, nhóm sẽ nghiên cứu bằng cách tích hợp thêm nhiều nguồn dữ liệu kinh tế và tài chính khác. Chúng em cũng sẽ thử nghiệm các phương pháp hiện đại như mạng nơ-ron sâu (Deep Neural Networks), hoặc mô hình học tăng cường (Reinforcement Learning) để cải thiện khả năng dự đoán trong các điều kiện biến động phức tạp hơn. Ngoài ra, chúng tôi kỳ vọng có thể phát triển một ứng dụng thực tế dựa trên mô hình hiện tại, cho phép người dùng theo dõi xu hướng giá vàng theo thời gian thực và nhận các khuyến nghị đầu tư phù hợp. Thông qua những bước tiến này, chúng tôi hy vọng đề tài sẽ trở thành một tiền đề vững chắc cho các nghiên cứu tương lai và tạo ra giá trị thiết thực trong việc hỗ trợ các quyết định đầu tư chiến lược.

# TÀI LIỆU THAM KHẢO

[1]. TS. Nguyễn Mạnh Cường, Slides bài giảng học phần *Phân tích dữ liệu lớn*, Trường Đại học Công Nghiệp Hà Nội.

[2]. BRANDS LOGOS, https://brandslogos.com/p/python-logo/. [Ngày truy cập: 15-11-2024]

[3]. *ChatGPT, Giải thích về mạng Long Short Term Memory (LSTM), https://chatgpt.com/c/67550f82-b3b4-800e-9cfe-ed725ac8fbb7*. [Ngày truy cập: 28-11-2024]

[4]. Phạm Đình Khánh, https://github.com/phamdinhkhanh/LSTM/blob/master/LSTM.png. [Ngày truy cập: 28-11-2024]