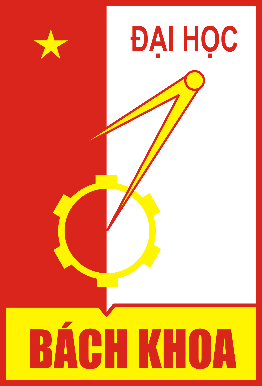
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN TRUYỀN THÔNG**



**BÁO CÁO NHẬP MÔN HỌC MÁY VÀ KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: Dự đoán giá bitcoin**

**Nhóm 11 - LỚP 157523**

**Giáo viên hướng dẫn: Nguyễn Thị Kim Anh**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thành viên nhóm** | MSSV | **Công việc** | **Đánh giá** |
| Tăng Minh Vũ | 20215519 | Xây dựng và huấn luyện mô hình | 20% |
| Vũ Văn Tú | 20215504 | Thu thập, tiền xử lý dữ liệu | 20% |
| Nguyễn Đức Hoàng | 20210377 | Phân tích dữ liệu & trực quan hóa | 20% |
| Lê Minh Hiếu | 20215368 | Xây dựng và huấn luyện mô hình | 20% |
| Nguyễn Quang Huy | 20225008 | Xây dựng mô hình, viết báo cáo | 20% |

**Hà Nội – 2025**

Mục lục

[1. Giới thiệu 4](#_Toc199056298)

[1.1 Lí do chọn đề tài 4](#_Toc199056299)

[1.2 Mô tả đề tài 4](#_Toc199056300)

[2. Tiền xử lý dữ liệu 5](#_Toc199056301)

[2.1. Phân tích dữ liệu khám phá , làm sạch dữ liệu và thêm các đặc trưng vào data\_frame 5](#_Toc199056302)

[2.2. Chuẩn bị dữ liệu cho mô hình học máy 6](#_Toc199056303)

[3. Mô hình đề xuất 9](#_Toc199056304)

[3.1 Mô hình Linear Regression 9](#_Toc199056305)

[3.2 Ridge Regression 11](#_Toc199056306)

[3.3 K-Nearest Neighbors (KNN) 12](#_Toc199056307)

[3.4 Random Forest 12](#_Toc199056308)

[3.5 Mô hình Support Vector Regression (SVR) 14](#_Toc199056309)

[3.6 Tổng quan 15](#_Toc199056310)

[4. Xây dựng chương trình 17](#_Toc199056311)

[4.1 Các thư viện đã được sử dụng 17](#_Toc199056312)

[4.2 Tải và load dữ liệu 18](#_Toc199056313)

[4.3 Tổng hợp dữ liệu theo giờ 18](#_Toc199056314)

[4.4 Thêm chỉ báo kỹ thuật 19](#_Toc199056315)

[4.5 Chuẩn bị dữ liệu cho huấn luyện mô hình 20](#_Toc199056316)

[4.6. Trực quan hóa kết quả phân loại (với 2 chiều) 20](#_Toc199056317)

[5. Các vấn đề gặp phải và cách giải quyết 21](#_Toc199056318)

[6. Các khám phá mới hoặc kết luận 22](#_Toc199056319)

# 1. Giới thiệu

## 1.1 Lí do chọn đề tài

Trong những năm gần đây, tiền mã hóa – đặc biệt là Bitcoin – đã trở thành một chủ đề thu hút sự quan tâm lớn trên toàn cầu, cả trong giới đầu tư lẫn lĩnh vực công nghệ. Giá trị của Bitcoin biến động mạnh, chịu ảnh hưởng từ nhiều yếu tố như tin tức, chính sách, cung cầu thị trường và các yếu tố kỹ thuật. Việc dự đoán xu hướng giá Bitcoin không chỉ có giá trị thực tiễn cao mà còn là một bài toán hấp dẫn trong lĩnh vực học máy và khai phá dữ liệu, nơi các mô hình có thể học từ dữ liệu quá khứ để đưa ra dự đoán cho tương lai.

Chọn đề tài này giúp người thực hiện có cơ hội áp dụng các kiến thức nền tảng của học máy vào một bài toán thực tế, rèn luyện kỹ năng xử lý dữ liệu thời gian, đánh giá mô hình và đưa ra nhận định khách quan.

## Mô tả đề tài

Đề tài tập trung vào việc dự đoán giá Bitcoin trong ngắn hạn (theo ngày hoặc theo giờ) dựa trên dữ liệu lịch sử. Dữ liệu được thu thập từ các nguồn công khai như sàn giao dịch tiền điện tử hoặc API tài chính. Các đặc trưng đầu vào bao gồm:

* Timestamp: Mốc thời gian
* Open: Giá mở cửa
* High: Giá cao nhất
* Low: Giá thấp nhất
* Close: Giá đóng cửa
* Volume: Khối lượng giao dịch

Sau khi xử lý và phân tích dữ liệu, nhóm sẽ áp dụng một số mô hình học máy như hồi quy tuyến tính (Linear Regression), rừng ngẫu nhiên (Random Forest), hoặc mạng nơ-ron nhân tạo (ANN) để huấn luyện và đánh giá khả năng dự đoán xu hướng giá Bitcoin. Kết quả mô hình sẽ được phân tích thông qua các chỉ số như RMSE, MAE, hoặc độ chính xác nếu chuyển thành bài toán phân loại (tăng / giảm).  
Mục tiêu của đề tài là xây dựng một hệ thống đơn giản có thể dự đoán xu hướng giá Bitcoin và rút ra nhận định về tính khả thi của việc ứng dụng học máy trong lĩnh vực tài chính.

# 2. Tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu đóng vai trò then chốt trong việc đảm bảo chất lượng của dữ liệu đầu vào cho các mô hình học máy, từ đó ảnh hưởng trực tiếp đến hiệu suất của các mô hình. Trong nghiên cứu này, dữ liệu gốc từ file CSV chứa dữ liệu giá Bitcoin được xử lý và phân tích dữ liệu khám phá (EDA) như sau:

## 2.1. Phân tích dữ liệu khám phá , làm sạch dữ liệu và thêm các đặc trưng vào data\_frame

* Đọc dữ liệu: Dữ liệu được đọc từ file btcusd\_1-min\_data.csv chứa thông tin giá Bitcoin theo từng phút.
* Chuyển đổi thời gian**:** Cột Timestamp ban đầu ở định dạng UNIX được chuyển sang định dạng datetime để thuận tiện cho việc xử lý chuỗi thời gian. Đồng thời, cột này được đặt làm chỉ số thời gian (index) của DataFrame.
* Loại bỏ giá trị thiếu (Missing Values): Các dòng chứa giá trị thiếu (NaN) được loại bỏ hoàn toàn khỏi tập dữ liệu bằng phương thức dropna().
* Tổng hợp dữ liệu theo khung giờ: Dữ liệu gốc theo phút được lấy mẫu lại theo khung giờ (1 giờ) bằng phương thức resample('H'), áp dụng các phép tổng hợp phù hợp:
  + Giá mở cửa (Open): Giá đầu tiên trong mỗi giờ.
  + Giá cao nhất (High): Giá cao nhất trong mỗi giờ.
  + Giá thấp nhất (Low): Giá thấp nhất trong mỗi giờ.
  + Giá đóng cửa (Close): Giá cuối cùng trong mỗi giờ.
  + Khối lượng (Volume): Tổng khối lượng giao dịch trong mỗi giờ.
* Tính toán các đặc trưng kỹ thuật: Dựa trên dữ liệu đã tổng hợp, một số đặc trưng kỹ thuật được tính toán nhằm hỗ trợ các mô hình học máy hiểu sâu hơn về xu hướng giá:
  + Đường trung bình động (Moving Average): MA6, MA12, MA24 là đường trung bình cộng giá đóng cửa trong 6 giờ gần nhất, giúp làm mượt dữ liệu và xác định xu hướng ngắn hạn.

A white background with text

AI-generated content may be incorrect.

* + Chỉ báo RSI (Relative Strength Index): đo độ mạnh/yếu của giá so với lịch sử tăng giảm gần đây, dao động từ 0 đến 100.

A computer screen shot of text

AI-generated content may be incorrect.

* + MACD và MACD Signal: Dùng để phát hiện tín hiệu mua/bán thông qua độ lệch giữa hai đường EMA.
    - MACD > Signal → Tín hiệu mua
    - MACD < Signal → Tín hiệu bán
  + Dải Bollinger (Bollinger Bands): Bao gồm BB\_Upper, BB\_Lower, và BB\_Width giúp đo độ biến động quanh một đường trung bình. Khi giá gần dải trên → có thể sắp giảm, gần dải dưới → có thể sắp tăng.
  + Biến động (Volatility) và tỷ lệ thay đổi giá (ROC6). Cho biết độ biến động trong một giờ, tính bằng khoảng cách từ đỉnh đến đáy. Volatility=High−Low
  + Làm sạch lần cuối: Sau khi tính toán các đặc trưng và biến mục tiêu, những dòng chứa giá trị NaN phát sinh từ các thao tác rolling hoặc shift tiếp tục được loại bỏ.

## 2.2. Chuẩn bị dữ liệu cho mô hình học máy

Dữ liệu đã làm sạch được tiếp tục xử lý để chuẩn bị cho việc huấn luyện mô hình như sau:

* Tách đặc trưng và nhãn:
  + Tập đặc trưng X: Bao gồm tất cả các đặc trưng kỹ thuật và đặc trưng thời gian.
  + Tập mục tiêu y: Gồm một trong các biến Next\_ih\_Price (ví dụ: Next\_1h\_Price) tùy thuộc vào bài toán dự đoán giá sau i giờ.
* Chuẩn hóa đặc trưng (Feature Scaling):
  + Các đặc trưng trong X và mục tiêu y được chuẩn hóa về khoảng [0, 1] bằng bộ biến đổi MinMaxScaler từ thư viện Scikit-learn.
  + Việc chuẩn hóa giúp tăng tốc độ hội tụ của các mô hình và đảm bảo các đặc trưng có quy mô tương đồng.
* Phân chia tập dữ liệu (Train-Test Split):
  + Tập huấn luyện (Training Set): Chiếm 80% dữ liệu đầu tiên theo trật tự thời gian, dùng để huấn luyện mô hình.
  + Tập kiểm tra (Test Set): Chiếm 20% dữ liệu cuối cùng, dùng để đánh giá độ chính xác của mô hình.
  + Việc chia tập được thực hiện theo thứ tự thời gian, không xáo trộn dữ liệu nhằm tránh rò rỉ thông tin giữa các tập.

A red and blue squares

AI-generated content may be incorrect.

A graph with blue lines and text

AI-generated content may be incorrect.

# 3. Mô hình đề xuất

Lựa chọn các chỉ số đánh giá mô hình hồi quy

* RMSE – Root Mean Squared Error: độ lệch trung bình bình phương
* MAE – Mean Absolute Error: Độ lệch trung bình tuyệt đối
* R² Score: Mức độ giải thích của mô hình (0 → 1).\
* MAPE – Mean Absolute Percentage Error: % sai số trung bình.

Sau khi hoàn tất việc chuẩn bị dữ liệu, nhóm tiến hành áp dụng một số thuật toán học máy để dự đoán phân lớp chỉ số tài chính dựa trên các đặc trưng đã được trích xuất. Các thuật toán được chọn lựa dựa trên tính chất của bài toán phân loại và kinh nghiệm từ các nghiên cứu tương tự.

## 3.1 Mô hình Linear Regression

Linear Regression là một trong những thuật toán học máy nền tảng và dễ hiểu nhất, thường được sử dụng trong các bài toán dự đoán giá trị liên tục. Nguyên lý hoạt động của mô hình là tìm ra mối quan hệ tuyến tính giữa biến đầu ra (biến mục tiêu) và một hoặc nhiều biến đầu vào (các đặc trưng). Cụ thể, mô hình cố gắng xác định một đường thẳng (hoặc siêu phẳng trong không gian nhiều chiều) sao cho tổng bình phương sai số giữa giá trị thực tế và giá trị dự đoán là nhỏ nhất — tức là mô hình tối thiểu hóa hàm mất mát theo phương pháp bình phương tối tiểu (Least Squares).

Trong bài toán này, Linear Regression được áp dụng để dự đoán giá Bitcoin trong một giờ tiếp theo, dựa trên các đặc trưng kỹ thuật được trích xuất từ dữ liệu lịch sử (ví dụ như giá mở cửa, giá đóng cửa, khối lượng giao dịch, chỉ báo RSI, MACD, v.v.). Việc lựa chọn Linear Regression là nhằm tạo ra một mô hình cơ bản, làm điểm khởi đầu để so sánh với các mô hình phi tuyến phức tạp hơn về sau như XGBoost, LSTM hoặc các mạng nơ-ron sâu.

A white background with black text

AI-generated content may be incorrect.

Kết quả thí nghiệm mô hình

* RMSE: $321.63
* MAE: $157.52
* R²: 0.9998
* MAPE: 0.3705%

Đánh giá tổng quan:

Kết quả trên cho thấy mô hình Linear Regression có khả năng dự đoán giá Bitcoin khá tốt trong ngắn hạn (một giờ tới) điều này là nhờ dữ liệu đầu vào tương tối phù hợp với mô hình này vỡi dữ liệu đầu vào là 1 bảng gồm các đặc trưng riêng biệt và các dữ liệu hầu hết đều là đều dạng số thực và đã loại bỏ hầu hết các dữ liệu lỗi trống cho nên ta có kết quả tương đối tốt. Với R² gần tuyệt đối và MAPE dưới 0.5%, mô hình này dù đơn giản nhưng mang lại hiệu suất ấn tượng trong bối cảnh dữ liệu đã được xử lý kỹ và đặc trưng hóa đầy đủ. Tuy nhiên, cần lưu ý rằng Linear Regression giả định mối quan hệ tuyến tính giữa các đặc trưng và biến mục tiêu — điều này có thể không còn đúng trong bối cảnh biến động mạnh của thị trường tiền mã hóa. Vì vậy, việc kết hợp hoặc thay thế bằng các mô hình phi tuyến khác sẽ được cân nhắc trong các phần tiếp theo nhằm cải thiện hơn nữa hiệu quả dự đoán.

## 3.2 Ridge Regression

Ridge Regression là một phiên bản cải tiến của Linear Regression, được bổ sung thêm một thành phần regularization L2 trong hàm mất mát để xử lý hiện tượng đa cộng tuyến giữa các biến đầu vào. Thay vì chỉ tối thiểu hóa tổng bình phương sai số, Ridge Regression thêm một hệ số phạt vào bình phương trọng số mô hình, giúp ngăn chặn việc hệ số hồi quy trở nên quá lớn, từ đó giảm nguy cơ overfitting.

Trong bài toán này, Ridge Regression được áp dụng nhằm kiểm soát độ phức tạp của mô hình trong bối cảnh dữ liệu có thể chứa nhiều đặc trưng tương quan cao như các chỉ báo kỹ thuật (RSI, EMA, MACD,…). Việc thêm regularization giúp mô hình ổn định hơn so với Linear Regression, đặc biệt khi số lượng đặc trưng tăng lên hoặc dữ liệu có nhiễu.

Kết quả thực nghiệm của mô hình Ridge Regression:

* RMSE: 320.04
* MAE: 156.85
* R²: 0.9998
* MAPE: 0.3671%

Đánh giá tổng quan:

So với Linear Regression, Ridge Regression mang lại hiệu quả tương đương nhưng có khả năng tổng quát hóa tốt hơn, đặc biệt khi số lượng đặc trưng tăng. Regularization giúp giảm overfitting và đảm bảo độ ổn định trong các trường hợp biến động dữ liệu mạnh. Đây là lựa chọn tốt cho các bài toán hồi quy có nhiều biến đầu vào tương quan cao, mặc dù cải thiện không quá rõ rệt nếu dữ liệu đã được xử lý kỹ.

## 3.3 K-Nearest Neighbors (KNN)

K-nearest neighbor (KNN) là một trong những thuật toán học có giám sát đơn giản nhất. Khi huấn luyện, thuật toán này gần như không học một điều gì từ dữ liệu huấn luyện mà ghi nhớ lại một cách máy móc toàn bộ dữ liệu đó. Mọi tính toán được thực hiện tại pha kiểm tra. KNN có thể được áp dụng vào các bài toán phân loại và hồi quy. KNN còn được gọi là một thuật toán lười học, instance-based [AKA91], hoặc memory-based learning.

KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới chỉ dựa trên thông tin của K điểm dữ liệu gần nhất trong tập huấn luyện, Khi có dữ liệu mới x, mô hình sẽ tìm ra tập hợp k điểm gần nhất với x, sau đó dựa vào các giá trị đầu ra tương ứng để dự đoán đầu ra của x.

Để xác định các điểm gần nhất, thuật toán cần sử dụng một hàm đo độ tương đồng giữa các điểm. Khoảng cách phổ biến nhất là: Khoảng cách Euclidean, Khoảng cách Manhattan, Khoảng cách Minkowski.

Với đầu vào dữ liệu có số chiều không quá lớn cùng dữ liệu rời rạc đây cũng là một mô hình đáng xem xét để thử huấn luyện tuy nhiên với dữ liệu đầu vào có khá nhiều điểm ngoại lệ với sự gia tăng đột ngột về giá, các điểm outlier cho nên kết quả của mô hình này không được quá tốt.

* RMSE: $8285.57
* MAE: $4057.83
* R²: 0.8571
* MAPE: 8.7585%

## 3.4 Random Forest

Random Forest là một mô hình học máy tổ hợp (ensemble learning), được xây dựng bằng cách kết hợp nhiều cây quyết định (Decision Tree) huấn luyện trên các tập dữ liệu con khác nhau. Kỹ thuật này giúp giảm thiểu hiện tượng overfitting và cải thiện độ chính xác bằng cách khai thác sự “đa dạng” giữa các mô hình thành phần. Trong bài toán hồi quy, Random Forest đưa ra dự đoán bằng cách trung bình hóa kết quả của tất cả các cây trong rừng.

A diagram of a process

Description automatically generated, Picture

Điểm mạnh của Random Forest nằm ở khả năng xử lý mối quan hệ phi tuyến phức tạp giữa các đặc trưng mà không cần giả định tuyến tính như Linear Regression. Ngoài ra, mô hình này cũng khá ổn định trước nhiễu và có thể xử lý tốt tập dữ liệu có nhiều đặc trưng kỹ thuật như trong bài toán dự đoán giá Bitcoin.

Trong dự án này, Random Forest được áp dụng để dự đoán giá Bitcoin trong một giờ tiếp theo, dựa trên các chỉ báo kỹ thuật đã được tính toán như MA, RSI, MACD, Bollinger Bands, ROC, Volatility, cũng như đặc trưng thời gian (giờ, thứ). Đây là mô hình phù hợp với bài toán hồi quy có tính phi tuyến cao và dữ liệu biến động mạnh.

Random Forest là một mô hình tổ hợp mạnh mẽ, tuy nhiên hiệu quả thực tế của nó phụ thuộc nhiều vào việc lựa chọn các siêu tham số phù hợp. Một số siêu tham số quan trọng ảnh hưởng đến hiệu suất mô hình bao gồm:

* n\_estimators: Số lượng cây trong rừng. Số lượng càng lớn thường giúp mô hình ổn định hơn, nhưng sẽ tăng chi phí tính toán.
* max\_depth: Độ sâu tối đa của mỗi cây. Độ sâu quá lớn dễ gây overfitting, trong khi quá nhỏ có thể dẫn đến underfitting.
* min\_samples\_split: Số lượng mẫu tối thiểu để chia một nút trong cây. Giá trị cao hơn giúp giảm độ phức tạp và chống overfitting.

Để tìm tổ hợp tốt nhất của các siêu tham số này, kỹ thuật tìm kiếm lưới (Grid Search) được sử dụng kết hợp với cross-validation (cv=3)

Sau khi tìm được tổ hợp siêu tham số tốt nhất, mô hình Random Forest được tái huấn luyện với cấu hình này và đánh giá lại hiệu suất trên tập kiểm tra. Ngoài ra, mô hình cũng trích xuất được tầm quan trọng của các đặc trưng đầu vào (feature importance), từ đó có thể xác định được những yếu tố nào ảnh hưởng mạnh mẽ nhất đến giá Bitcoin trong dự báo.

Đánh giá tổng quan:

* RMSE: $7400.27
* MAE: $2284.47
* R²: 0.8860
* MAPE: 3.3339%

Mặc dù Random Forest được kỳ vọng là một trong những mô hình mạnh mẽ nhất nhờ vào khả năng học phi tuyến và chống overfitting, nhưng trong dự án này, kết quả thực nghiệm cho thấy hiệu suất của mô hình chưa đạt kỳ vọng. Nguyên nhân có thể đến từ việc đặc điểm chuỗi thời gian chưa được khai thác đầu, hoặc mô hình chưa được tối ưu siêu tham số (ví dụ: số lượng cây, độ sâu tối đa, min\_samples split,...).

Tuy vậy, Random Forest vẫn là một mô hình tiềm năng nếu tiếp tục được tinh chỉnh, và đặc biệt có ích trong việc đánh giá tầm quan trọng của từng đặc trưng, giúp hiểu rõ hơn yếu tố nào ảnh hưởng nhiều nhất đến giá Bitcoin.

## 3.5 Mô hình Support Vector Regression (SVR)

Support Vector Regression (SVR) là một biến thể của Support Vector Machine (SVM) được dùng cho bài toán hồi quy. Mục tiêu của SVR là tìm một siêu phẳng hồi quy sao cho tất cả các điểm dữ liệu nằm trong khoảng cách ε (epsilon) với sai số nhỏ nhất có thể, đồng thời giữ cho siêu phẳng đó có độ phức tạp thấp nhất (theo chuẩn L2 của các hệ số). SVR có thể mô hình hóa các quan hệ phi tuyến thông qua các hàm kernel như RBF, Polynomial,…

Trong bài toán dự đoán giá Bitcoin, SVR được kỳ vọng là mô hình học phi tuyến mạnh mẽ, có khả năng nắm bắt các xu hướng khó lường trong dữ liệu tài chính. Mặc dù hiệu quả của SVR phụ thuộc lớn vào việc lựa chọn các siêu tham số như C (độ phạt), ε (ngưỡng sai số chấp nhận), và kernel, nhưng đây là mô hình rất tiềm năng khi dữ liệu có cấu trúc phi tuyến ổn định.

Kết quả thực nghiệm của mô hình SVR:

* RMSE (Root Mean Squared Error): $11978.55

Đây là mức sai số lớn nhất, thể hiện mô hình dự đoán lệch khá xa thực tế.

* MAE (Mean Absolute Error): $5660.74

Sai số tuyệt đối cao, cho thấy độ ổn định không đảm bảo.

* R² (Hệ số xác định): 0.7012

Mức giải thích thấp, chỉ khoảng 70% phương sai được mô hình hóa.

* MAPE (Mean Absolute Percentage Error): 9.8339%

Sai số phần trăm cao nhất trong tất cả mô hình, không phù hợp với yêu cầu độ chính xác cao.

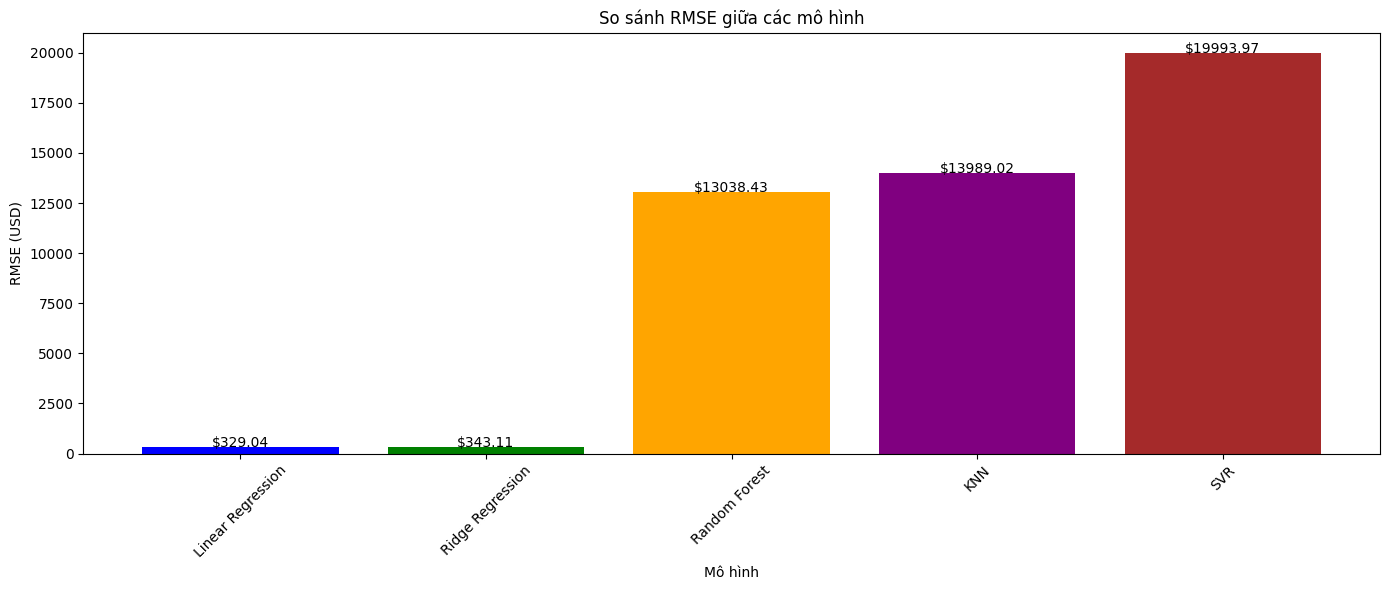
Đánh giá tổng quan:

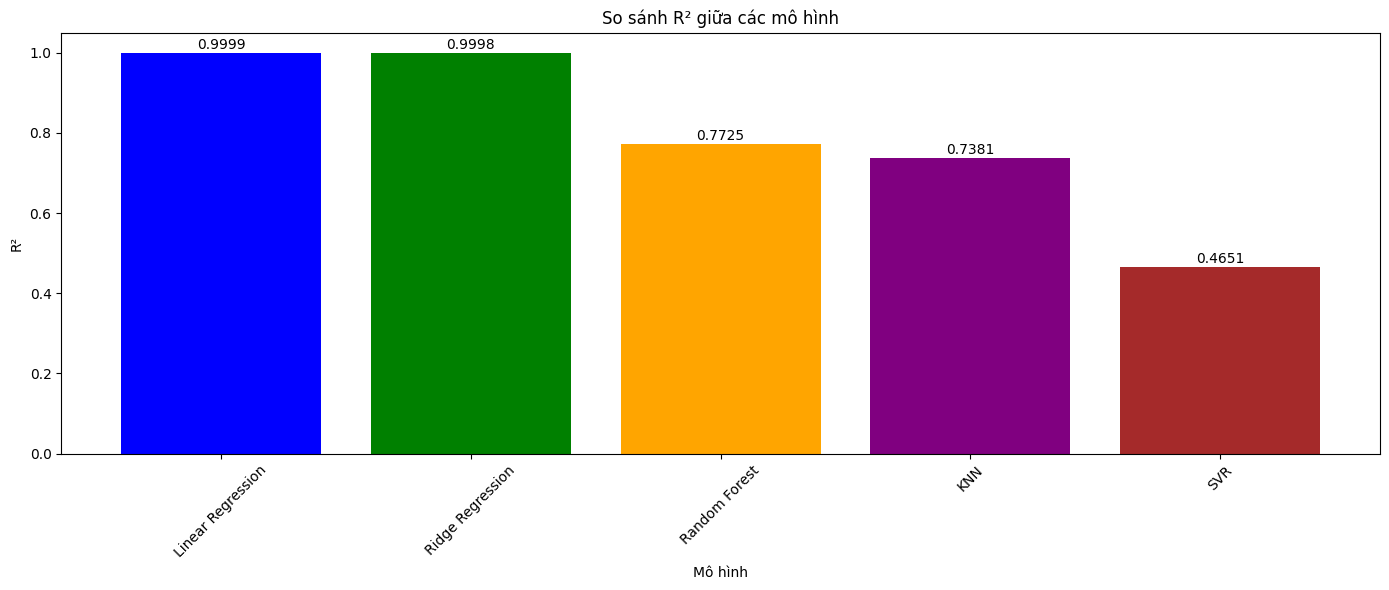
Dù SVR là một mô hình mạnh mẽ trên lý thuyết, nhưng trong thực nghiệm với dữ liệu dự đoán giá Bitcoin, kết quả không như mong đợi. Lý do có thể đến từ việc chưa tối ưu hóa các siêu tham số như C, ε và gamma (kernel), hoặc dữ liệu biến động quá mạnh, vượt ngoài khả năng mô hình hóa của SVR trong cấu hình hiện tại.

Nếu được áp dụng kỹ thuật tối ưu hóa tham số như GridSearchCV, kết hợp với feature selection hoặc giảm chiều dữ liệu đầu vào, SVR có thể cải thiện hiệu quả đáng kể. Tuy nhiên, với cấu hình mặc định, SVR chưa đủ tốt để ứng dụng trong dự đoán tài chính thực tế.

## 3.6 Tổng quan

Qua thí nghiệm với 5 mô hình trên nhóm e nhận thấy phương pháp phù hợp nhất là linear regression hoặc ridge Regression. Nguyên nhân cho kết luận này là vì dữ liệu đầu vào độc lập với cái giá trị rời rạc và đầy đủ điều này dẫn đến phù hợp để áp dụng phương pháp này cho bài toán này.





# 4. Xây dựng chương trình

## 4.1 Các thư viện đã được sử dụng

A screenshot of a computer code

AI-generated content may be incorrect.

Thư viện numpy: là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.

Thư viện pandas: là thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác và phân tích dữ liệu. Đây là bộ công cụ mạnh mẽ giúp xử lý dữ liệu dưới dạng bảng (DataFrame) dễ dàng và linh hoạt.

Thư viện matplotlib: được sử dụng để tạo biểu đồ từ dữ liệu. Giúp trực quan hóa dữ liệu để đánh giá và phân tích kết quả một cách tốt hơn.

Thư viện seaborn: là thư viện mở rộng từ matplotlib, giúp tạo biểu đồ thống kê với giao diện đẹp mắt và dễ sử dụng, phù hợp cho phân tích dữ liệu nâng cao.

Thư viện sklearn (Scikit-learn): là thư viện mạnh mẽ dành cho các thuật toán học máy viết bằng Python. Hỗ trợ chia tập dữ liệu, chuẩn hóa dữ liệu và đánh giá hiệu năng mô hình bằng các chỉ số như mean squared error hay r2 score.

Thư viện datetime: hỗ trợ xử lý dữ liệu ngày giờ, giúp thao tác, chuyển đổi và tính toán với dữ liệu thời gian một cách dễ dàng.

Thư viện warnings: dùng để quản lý các cảnh báo khi chạy chương trình. Thường được dùng để tắt các cảnh báo không quan trọng giúp output gọn hơn.

## 4.2 Tải và load dữ liệu

A screenshot of a computer program

AI-generated content may be incorrect.

## 4.3 Tổng hợp dữ liệu theo giờ

A white background with green text

AI-generated content may be incorrect.

## 4.4 Thêm chỉ báo kỹ thuật

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

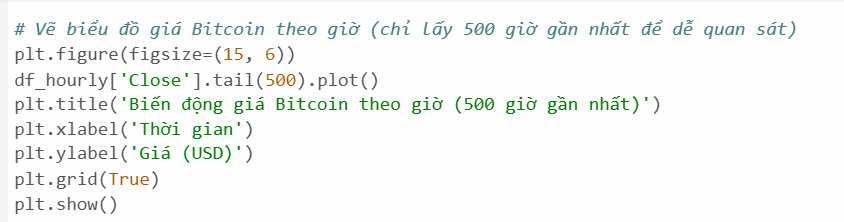
AI-generated content may be incorrect.

## 4.5 Chuẩn bị dữ liệu cho huấn luyện mô hình

A screenshot of a computer program

AI-generated content may be incorrect.

## 4.6. Trực quan hóa kết quả phân loại (với 2 chiều)



plt.scatter(...): Vẽ biểu đồ phân tán (scatter plot) với:

* Trục X là đặc trưng đầu tiên.
* Trục Y là đặc trưng thứ hai.
* Màu sắc thể hiện nhãn dự đoán (y\_pred).
* cmap='viridis': Dùng bảng màu viridis để phân biệt các nhãn.plt.title, xlabel, ylabel, colorbar: Đặt tiêu đề, nhãn trục và chú giải màu.
* plt.show(): Hiển thị biểu đồ.

# 5. Các vấn đề gặp phải và cách giải quyết

5.1. Khó khăn trong việc thu thập và xử lý dữ liệu

*Vấn đề:* Dữ liệu thực tế từ thị trường Bitcoin biến động liên tục theo thời gian, kích thước lớn, chứa nhiều nhiễu và không đầy đủ.

*Giải pháp:* Sử dụng API và nguồn dữ liệu đáng tin cậy (như Kaggle, Yahoo Finance). Áp dụng các kỹ thuật xử lý dữ liệu như loại bỏ giá trị thiếu, chuẩn hóa dữ liệu với MinMaxScaler, và phân tích thống kê ban đầu bằng pandas, seaborn.

5.2. Lựa chọn mô hình phù hợp

*Vấn đề:* Ban đầu chưa rõ mô hình nào hoạt động hiệu quả nhất với dữ liệu biến động theo thời gian như giá Bitcoin.

*Giải pháp:* Thử nghiệm nhiều mô hình học máy như Linear Regression, Ridge, Decision Tree, Random Forest, KNN, và SVM. Sử dụng kỹ thuật cross-validation và các chỉ số đánh giá như MAE, MSE, R2 Score để so sánh và lựa chọn mô hình tốt nhất.

5.3. Khó khăn trong việc tối ưu siêu tham số (hyperparameter tuning)

*Vấn đề:* Kết quả ban đầu chưa tốt do chưa điều chỉnh đúng siêu tham số cho từng mô hình.

*Giải pháp:* Sử dụng GridSearchCV trong thư viện Scikit-learn để tự động tìm bộ tham số tối ưu nhất giúp tăng hiệu suất dự đoán.

5.4. Hiểu và triển khai thư viện học máy

*Vấn đề:* Một số thành viên chưa quen với cách sử dụng các thư viện như scikit-learn, matplotlib, seaborn.

*Giải pháp:* Phân công thành viên học thêm tài liệu chính thức và khóa học online miễn phí (như Coursera, YouTube). Các thành viên hỗ trợ nhau trong việc debug và trình bày kết quả.

5.5. Vấn đề về trực quan hóa kết quả

*Vấn đề:* Khó khăn trong việc thể hiện kết quả dự đoán sao cho dễ hiểu và trực quan.

*Giải pháp:* Áp dụng các biểu đồ như line plot (so sánh giá thực tế và giá dự đoán), heatmap (phân tích tương quan), biểu đồ phân phối sai số bằng matplotlib và seaborn.

5.6. Quản lý thời gian và phân công công việc

*Vấn đề:* Do có nhiều môn học khác và thời gian hạn chế, nhóm gặp khó khăn trong việc sắp xếp tiến độ làm đồ án.

*Giải pháp:* Chia công việc cụ thể theo từng giai đoạn: thu thập dữ liệu, tiền xử lý, xây dựng mô hình, đánh giá và viết báo cáo. Tổ chức họp nhóm định kỳ để kiểm tra tiến độ và hỗ trợ lẫn nhau.

# 6. Các khám phá mới hoặc kết luận

Từ kết quả thực nghiệm, có thể rút ra một số kết luận:

Mô hình Linear Regression đã được chọn là mô hình tốt nhất cho dự đoán giá Bitcoin theo giờ với RMSE thấp nhất ($321.63) và R² cao nhất (0.9998).

Thị trường tiền điện tử rất biến động và chịu ảnh hưởng của nhiều yếu tố không lường trước được. Dự đoán này chỉ nên được sử dụng như một tham khảo, không nên dùng làm cơ sở duy nhất cho quyết định đầu tư.

Trong tương lai, hệ thống có thể được cải thiện bằng cách:

* Bổ sung thêm các đặc trưng từ phân tích cảm xúc trên mạng xã hội
* Thử nghiệm các mô hình học sâu như LSTM cho dự đoán chuỗi thời gian
* Phát triển hệ thống cảnh báo khi giá đạt ngưỡng nhất định
* Tích hợp dữ liệu từ nhiều sàn giao dịch để có cái nhìn toàn diện hơn