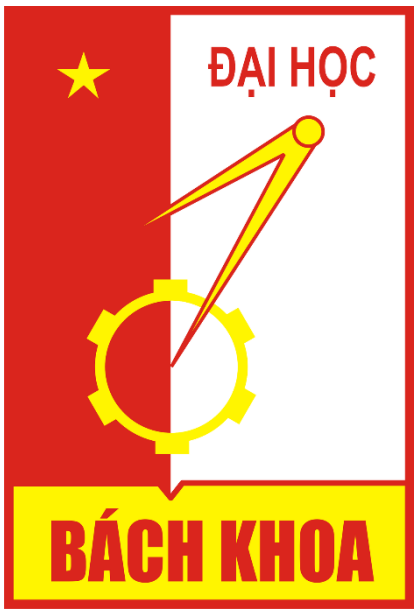
ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ Thông tin và Truyền thông



Project 2

Đề tài**: Ứng dụng Machine Learning trong bài toán Dự đoán giá Cổ phiếu**

Giảng viên hướng dẫn:	TS.Nguyễn Bá Ngọc	
Sinh viên thực hiện:	Nguyễn Ngọc Quỳnh Anh	20204631
	Trần Thị Anh	20204512

HÀ NỘI, 6/2023

Contents

1. Giới thiệu chung về đề tài

Bài toán dự đoán giá cổ phiếu là một trong những bài toán thuộc lĩnh vực tài chính và đầu tư. Giá cổ phiếu luôn biến động và có những thay đổi nhanh chóng bởi thị trường chứng khoán được đặc trưng bởi tính năng động, khó dự đoán và phi tuyến tính. Dự đoán giá cổ phiếu là một nhiệm vụ đầy thách thức vì nó phụ thuộc vào nhiều yếu tố bao gồm nhưng không giới hạn ở các điều kiện chính trị, kinh tế toàn cầu, báo cáo tài chính và hiệu suất của công ty, v.v

Việc dự đoán xu hướng giá cổ phiếu giúp các nhà đầu tư của các công ty có thể đưa ra quyết đinh thông minh hơn. Với mục tiêu của nó là dự đoán giá cổ phiếu của một công ty trong tương lai dựa trên những thông tin và dữ liệu có sẵn, bài toán này có thể được giải quyết bằng phương pháp học máy và khai phá dữ liệu. Tuy nhiên , việc dự đoán dựa trên các mô hình học máy luôn tồn tại một mức độ sai số và có tính rủi ro vì thị trường chứng khoán còn phụ thuộc vào rất nhiều yếu tố về kinh tế, xã hội, thị trường.

Mô tả bài toán: Cho trước một tập dữ liệu về giá cổ phiếu của một công ty trong quá khứ. Nhiệm vụ là sử dụng các phương pháp học máy để xây dựng mô hình dự đoán giá đóng cửa (Close Price) cổ phiếu của công ty đó trong tương lai. Đây là bài toán học có giám sát.

2. Lý thuyết về chứng khoán, cổ phiếu

Cổ phiếu là một loại giấy chứng nhận sở hữu cổ phần trong một công ty cổ phần. Khi một người mua cổ phiếu của một công ty, người đó trở thành một cổ đông của công ty đó và sở hữu một phần vốn của công ty tương ứng với số lượng cổ phiếu đã mua.

Cổ phiếu là một công cụ tài chính cho phép công ty huy động vốn từ nhà đầu tư để đầu tư vào hoạt động kinh doanh và mở rộng quy mô. Cổ phiếu cũng cung cấp cho chủ sở hữu (cổ đông) quyền hưởng lợi nhuận (nếu có) của công ty dưới dạng cổ tức hoặc tăng giá trị cổ phiếu.

Cổ phiếu thường giao dịch trên các sàn giao dịch chứng khoán, và giá trị của chúng thay đổi theo thời gian tùy thuộc vào tình hình tài chính và hoạt động kinh doanh của công ty, cũng như các yếu tố thị trường và kinh tế.

Việc đầu tư vào cổ phiếu mang theo cơ hội sinh lời lớn nhưng cũng đi kèm với nguy cơ rủi ro. Trong quá trình giao dịch cổ phiếu, giá có thể tăng lên hoặc giảm xuống, và việc lựa chọn cổ phiếu phù hợp cũng đòi hỏi sự nghiên cứu và hiểu biết về công ty, ngành nghề và thị trường tài chính.

Một số khái niệm :

- **Phiên giao dịch (Tradding session):** là khoảng thời gian diễn ra mọi hoạt động mua bán, trao đổi, đặt lệnh, hủy lệnh đầu tư trên các sàn giao dịch tham gia. Một phiên giao dịch có thể là 1 buổi hoặc 1 ngày hoặc 1 phiên khớp lệnh. Hiện nay, thời gian giao dịch của một ngày thường được bắt đầu lúc 9h sáng và kết thúc phụ thuộc vào các sàn giao dịch.

- **Khớp lệnh:** trong thị trường chứng khoán là việc thực hiện xong thoả thuận giữa bên mua và bên bán trên bảng giao dịch điện tử trực tuyến. Lệnh của các nhà đầu tư được ghép với nhau để giao dịch theo mức giá phù hợp với nguyên tắc ưu tiên khớp lệnh của thị trường. Hệ thống giao dịch khớp lệnh mua và

lệnh bán theo nguyên tắc ưu tiên về giá và thời gian, cụ thể: **Ưu tiên về giá**: Đối với lệnh mua, nếu mức giá cao hơn thì sẽ được ưu tiên thực hiện trước. Đối với lệnh bán, nếu mức giá thấp hơn thì sẽ được ưu tiên thực hiện trước. **Ưu tiên về thời gian**: Trong trường hợp các lệnh đang có cùng mức giá thì lệnh nhập vào hệ thống giao dịch trước sẽ được ưu tiên thực hiện trước.

- **Giá mở cửa (Opening Price):** được xác định bởi sự khớp lệnh giữa người mua và người bán trong khoảng thời gian ngắn nhất sau khi ngày giao dịch mới bắt đầu. Giá mở cửa quan trọng vì nó có thể ảnh hưởng đến xu hướng giá của tài sản trong suốt phiên giao dịch.
- **Giá cao nhất (High Price):** là mức giá cao nhất trong 1 phiên giao dịch hoặc trong 1 chu kỳ theo dõi biến động giá.
- **Giá thấp nhất (Low Price):** là mức giá thấp nhất trong 1 phiên giao dịch hoặc trong 1 chu kỳ theo dõi biến động giá.
- **Giá trần (Ceiling Price) :** là mức giá cao nhất trong ngày giao dịch mà các nhà đầu tư có thể đặt lệnh mua bán. Giá trần = Giá tham chiếu x (100% + Biên độ giao động)
- **Giá sàn (Floor Price):** là mức giá thấp nhất trong ngày giao dịch mà các nhà đầu tư có thể đặt lệnh mua bán. Giá sàn = Giá tham chiếu x (100% – Biên độ giao động)
- **Giá đóng cửa (Closing Price)** : là mức giá cuối cùng được chọn vào lúc đóng cửa phiên giao dịch trong ngày đồng thời sẽ là giá tham chiếu của cổ phiếu đó cho phiên giao dịch kế tiếp. Giá đóng cửa trên cổ phiếu được xem là một con số tiêu chuẩn được thực hiện theo dõi bởi các nhà đầu tư, tổ chức tài chính và các tổ chức khác đưa ra quyết định về cổ phiếu và công ty. Bởi thế giá đóng cửa được các nhà đầu tư, thương nhân, tổ chức tài chính, cơ quan quản lý và các bên liên quan khác sử dụng nó làm điểm tham chiếu để xác định hiệu suất

trong một thời gian nhất định như một năm, một tuần và trong khung thời gian ngắn hơn như một phút hoặc ít hơn. Giá đóng cửa của trên cổ phiếu là căn cứ để các nhà đầu tư xem xét, so sánh với giá sau thị trường của cổ phiếu để quyết định có nên đầu tư hay không.

- **Khối lượng giao dịch (Volume):** tổng số cổ phiếu được khớp lệnh trong phiên ngày hôm đó đối với một mã cổ phiếu hay một sàn chứng khoán cụ thể. Nó không chỉ thể hiện số lượng cổ phiếu được khớp lệnh trong phiên mà còn cho thấy nhu cầu giao dịch của nhà đầu tư, xu hướng giá và tiềm năng cổ phiếu trong thời gian sắp tới.

3.Cơ sở lý thuyết về các phương pháp học máy

3.1 Hồi quy tuyến tính – Linear Regression

3.1.1 Hồi quy tuyến tính là gì?

Hồi quy tuyến tính là một loại phân tích thống kê được sử dụng để dự đoán mối quan hệ giữa hai biến. Nó giả định mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc, và nhằm mục đích tìm ra đường thẳng phù hợp nhất để mô tả mối quan hệ. Đường thẳng được xác định bằng cách giảm thiểu tổng bình phương chênh lệch giữa giá trị dự đoán và giá trị thực.

Hồi quy tuyến tính thường được sử dụng trong nhiều lĩnh vực, bao gồm kinh tế, tài chính và khoa học xã hội, để phân tích và dự đoán xu hướng của dữ liệu. Nó cũng có thể được mở rộng cho hồi quy tuyến tính bội, trong đó có nhiều biến độc lập và hồi quy logistic, được sử dụng cho các bài toán phân loại nhị phân.

3.1.2 Thiết lập công thức

Hồi quy tuyến tính là phương pháp học có giám sát: cần học 1 hàm $y = f(x)$ từ một tập học cho trước $D = \{(x_1 , y_1), (x_2 , y_2),\ldots, (x_M, y_M)\}$ trong đó y_i xấp xỉ bằng $f(x_i)$ với mọi i:

- Mỗi quan sát được biểu diễn bằng một véctơ n chiều, chẳng hạn $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$
- Mỗi chiều biểu diễn một thuộc tính (attribute/feature)
- Giả thuyết hàm $y = f(x)$ là hàm có dạng tuyến tính:

$$f(x) = w_0 + w_1 x_1 + \ldots + w_n x_n$$

- Học một hàm hồi quy tuyến tính thì tương đương với việc học véctơ trọng số:

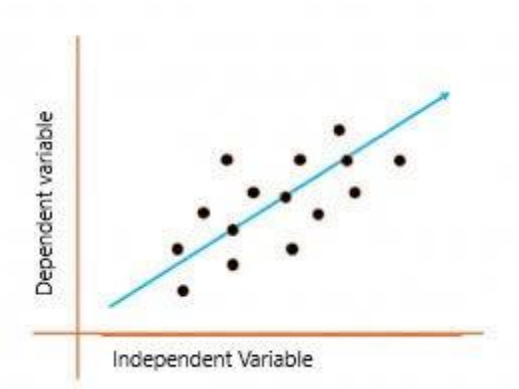
$$w = (w_0 , w_1 , \ldots, w_n)^T$$

- **Phán đoán cho quan sát tương lai**: $z = (z_1 , z_2 , \ldots, z_n)^T$

  Cần dự đoán giá trị đầu ra, bằng cách áp dụng hàm mục tiêu đã học được f:

$$f(z) = w_0 + w_1 x_1 + \ldots + w_n x_n$$

Hồi quy tuyến tính cho thấy mối quan hệ tuyến tính giữa biến độc lập (bộ dự báo) tức là trục X và biến phụ thuộc (đầu ra) tức là trục Y, được gọi là hồi quy tuyến tính. Nếu có một biến đầu vào duy nhất X (biến độc lập), hồi quy tuyến tính như vậy được gọi là hồi quy tuyến tính đơn giản.



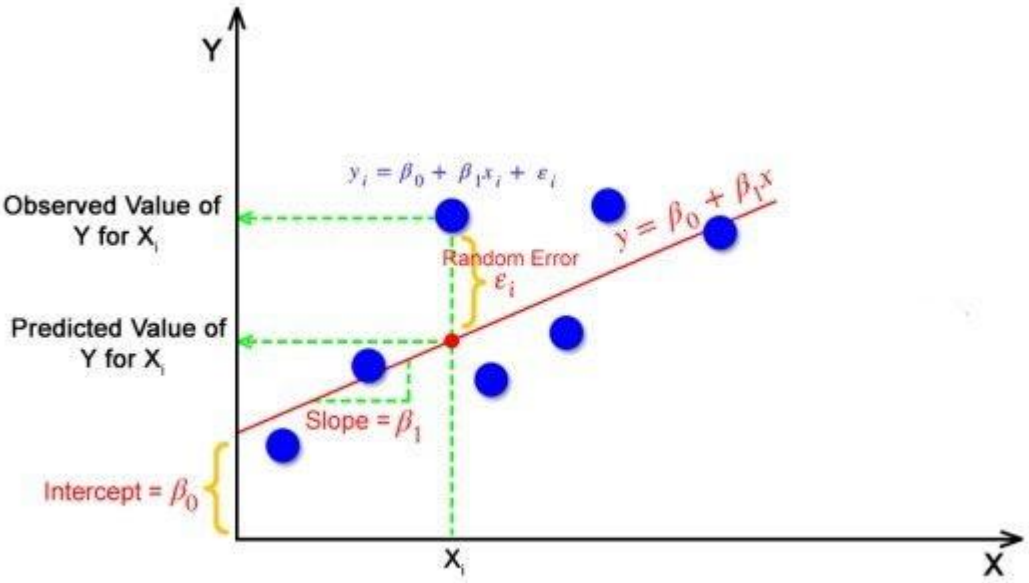
Để tính toán hồi quy tuyến tính để tìm ra đường phù hợp nhất, xem xét công thức sau:

$\mathbf{Y_i = \beta_0 + \beta_1 X_i}$

Trong đó:

- $\mathbf{Y_i}$ = Biến phụ thuộc
- $\boldsymbol{\beta_0}$: Hằng số
- $\boldsymbol{\beta_1}$: Hệ số góc
- $\mathbf{X_i}$: Biến độc lập

Thuật toán này giải thích mối quan hệ tuyến tính giữa biến phụ thuộc (đầu ra) y và biến độc lập (bộ dự báo) X bằng một đường thẳng $\mathbf{Y = \beta_0 + \beta_1 X}$



Mục tiêu của thuật toán hồi quy tuyến tính là lấy các giá trị tốt nhất cho β_0 và β_1 để tìm đường phù hợp nhất. Đường phù hợp nhất là đường có ít lỗi nhất, nghĩa là lỗi giữa giá trị dự đoán và giá trị thực tế phải ở mức tối thiểu.

Trong hồi quy, chênh lệch giữa giá trị quan sát được của biến phụ thuộc (yi) và giá trị dự đoán (predicted) được gọi là phần dư.

$\varepsilon_i = y_{predicted} - y_i$; $y_{predicted} = \beta_0 + \beta_1 X$

Vậy tìm hàm phù hợp nhất như thế nào?

⇨ Chúng ta cần 1 hàm để đánh giá là đường thẳng với bộ tham số (β_0 , β_1) hiện tại có tốt hay không.

Với mỗi điểm dữ liệu (x_i, y_i) độ chênh lệch giữa giá thật và giá dự đoán được tính bằng: $\frac{1}{2} * (\hat{y}_i - y_i)^2$.

Và độ chênh lệch trên toàn bộ dữ liệu tính bằng tổng chênh lệch của từng điểm:

J = $J = \frac{1}{2} * \frac{1}{N} * (\sum_{i=1}^{N} (\hat{y}_i - y_i)^2)$ (N là số điểm dữ liệu).

Nhận xét:

- J không âm
- J càng nhỏ thì đường thẳng càng gần điểm dữ liệu. Nếu J = 0 thì đường thẳng đi qua tất các điểm dữ liệu.

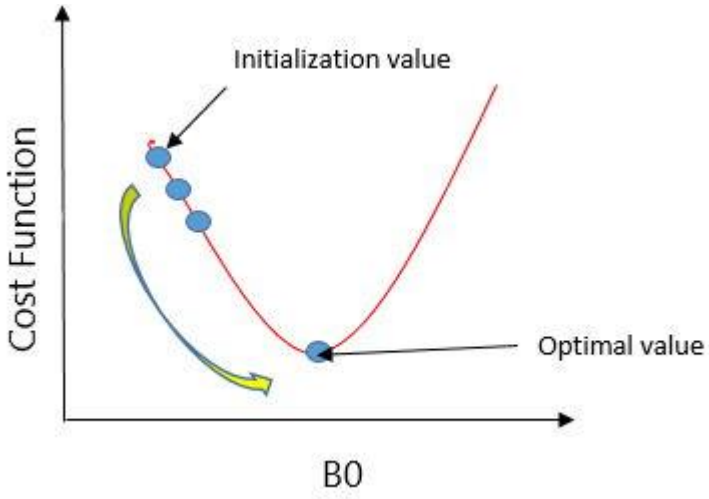
J được gọi là loss function, hàm để đánh giá xem bộ tham số hiện tại có tốt với dữ liệu không.

=> Bài toán tìm đường thẳng gần các điểm dữ liệu nhất trở thành tìm (β_0 , β_1) sao cho hàm J đạt giá trị nhỏ nhất.

Giờ cần một thuật toán để tìm giá trị nhỏ nhất của hàm J(β_0 , β_1). Đó chính là thuật toán gradient descent.

3.1.3 Thuật toán gradient descent

Gradient Descent là một trong những thuật toán tối ưu hóa hàm chi phí (hàm mục tiêu) để đạt được giải pháp tối ưu tối thiểu. Để tìm giải pháp tối ưu, chúng ta cần giảm loss function (Cost function) cho tất cả các điểm dữ liệu. Điều này được thực hiện bằng cách cập nhật lặp đi lặp lại các giá trị của β_0 và β_1 cho đến khi chúng tôi nhận được giải pháp tối ưu.

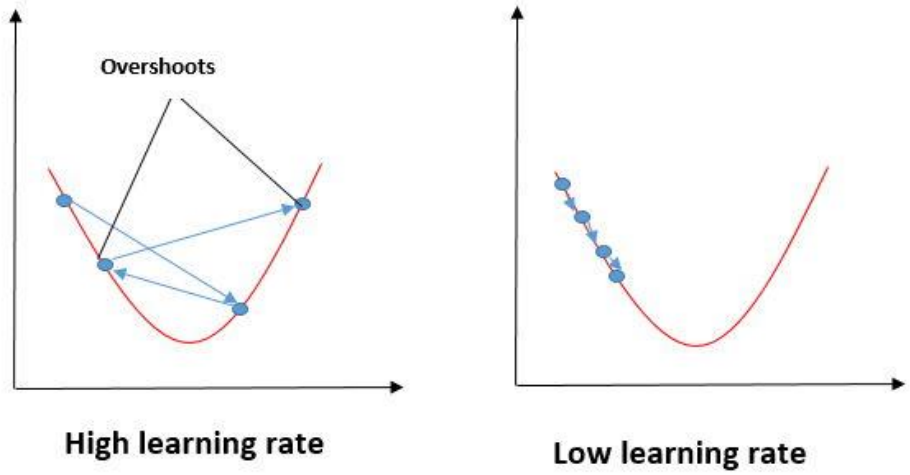


Gradient descent là thuật toán tìm giá trị nhỏ nhất của hàm số f(x) dựa trên đạo hàm.

Thuật toán:

1. Khởi tạo giá trị $x = x_0$ tùy ý
2. Gán x = x - learning_rate * f'(x) (learning_rate là hằng số dương ví dụ learning_rate = 0.001)
3. Tính lại f(x): Nếu f(x) đủ nhỏ thì dừng lại, ngược lại tiếp tục bước 2

Learning rate lớn và nhỏ:



3.1.4 Ưu điểm – Nhược điểm của Hồi quy tuyến tính

Ưu điểm:

- Đơn giản: Hồi quy tuyến tính là một phương pháp đơn giản và dễ hiểu, không đòi hỏi nhiều kiến thức chuyên sâu về thống kê để áp dụng.
- Hiệu quả tính toán: Công thức tính toán và ước lượng trong hồi quy tuyến tính rất nhanh chóng và hiệu quả, đặc biệt là đối với tập dữ liệu lớn.
- Tính diễn giải: Kết quả của hồi quy tuyến tính có thể dễ dàng diễn giải, giúp hiểu rõ hơn về mối quan hệ giữa biến phụ thuộc và biến độc lập.

Nhược điểm:

- Giả định về tuyến tính: Hồi quy tuyến tính giả định rằng mối quan hệ giữa biến phụ thuộc và biến độc lập là tuyến tính. Trong trường hợp mối quan hệ này không tuyến tính, kết quả của hồi quy tuyến tính có thể không chính xác hoặc không hợp lý.
- Độc lập tuyến tính: Hồi quy tuyến tính giả định rằng các biến độc lập là độc lập tuyến tính với nhau. Nếu có sự tương quan mạnh giữa các biến độc lập, điều này có thể dẫn đến việc suy biến sai, ước lượng không chính xác và kết quả không đáng tin cậy.
- Quan trọng giá trị ngoại lai: Hồi quy tuyến tính dễ bị ảnh hưởng bởi các giá trị ngoại lai trong dữ liệu. Các điểm dữ liệu cách biệt xa so với phân phối chung có thể làm sai lệch kết quả hồi quy.
- Không phù hợp với mối quan hệ phi tuyến: Hồi quy tuyến tính không thể mô hình hóa mối quan hệ phi tuyến giữa biến phụ thuộc và biến độc lập một cách chính xác. Trong trường hợp này, cần sử dụng các phương pháp hồi quy phi tuyến thích hợp hơn.

3.2 Rừng ngẫu nhiên – Random Forest

3.2.1 Rừng ngẫu nhiên là gì?

Random forests (RF) là một phương pháp dành cho phân lớp và hồi quy. Được đề xuất bởi Leo Breiman (2001).

Ý tưởng: là một sự kết hợp của các cây quyết định, bằng cách lấy trung bình các phán đoán của các cây.

- Mỗi cây trong đó là 1 cây đơn giản nhưng ngẫu nhiên.
- Mỗi cây được sinh ra phụ thuộc vào cách lựa chọn các thuộc tính trong quá trình học

3 thành phần cơ bản

- Ngẫu nhiên hoá và không cắt tỉa:
    - Đối với mỗi cây, tại mỗi nút ta chọn ngẫu nhiên một nhóm nhỏ các thuộc tính để chia.
    - Tính mốc chia tốt nhất, và phân nhánh cây.
    - Cây đó sẽ được sinh ra với cỡ lớn nhất, mà không dùng cắt tỉa.
- Tổng hợp: mỗi phán đoán về sau thu sau thu được bằng cách lấy trung bình các phán đoán từ tất cả các cây.
- Bagging: tập học dành cho mỗi cây được sinh ra bằng cách lấy ngẫu nhiên (có trùng lặp) từ tập học ban đầu.

Thuật toán

- Đầu vào: tập học D, số cây K
- Tạo K cây, mỗi cây được sinh ra như sau:
    - Xây dựng một tập con Di bằng cách lấy ngẫu nhiên (có trùng lặp) từ D.
    - Học cây thứ i từ Di như sau:

- Tại mỗi đỉnh (nút):
    - chọn ngẫu nhiên một tập con c p con các thuộc tính
    - phân nhánh cây dựa trên tập thuộc tính đó.
- Cây này sẽ được sinh ra với cỡ lớn nhất, không dùng cắt tỉa.

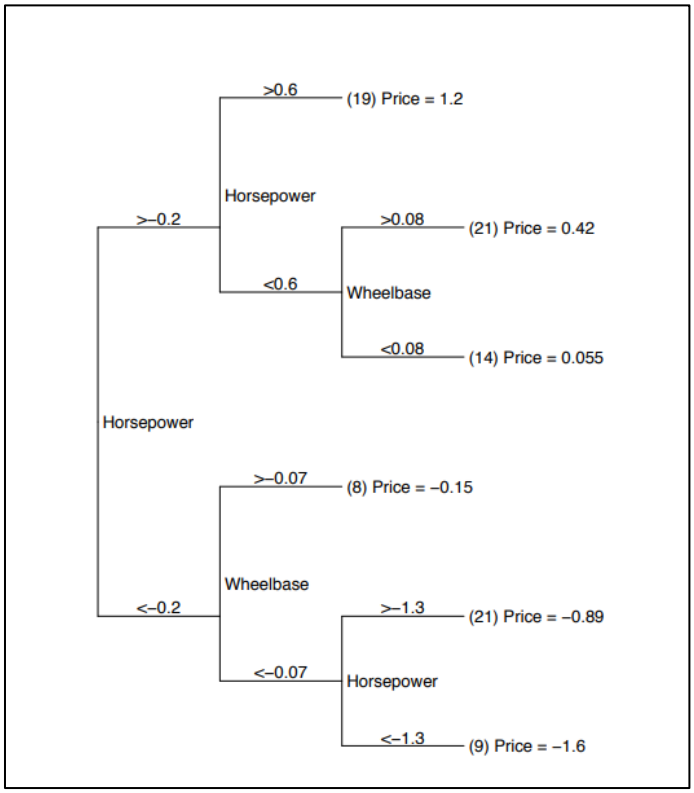
- Mỗi phán đoán về sau thu được bằng cách lấy trung bình các phán đoán từ tất cả các cây

3.2.2 Rừng ngẫu nhiên cho bài toán Hồi quy

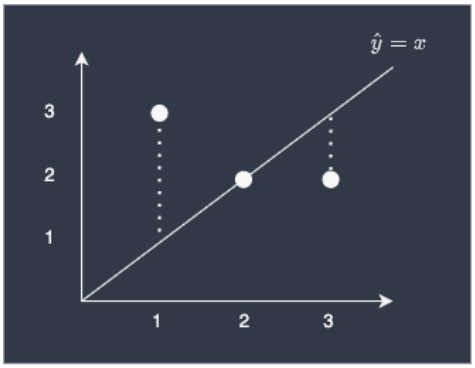
- **Cây hồi quy là gì?**

Cây hồi quy là cây quyết định trong đó các biến mục tiêu có thể nhận các giá trị liên tục thay vì các nhãn lớp ở dạng lá.

Ví dụ:



- Mean Square Error (Lỗi bình phương trung bình):

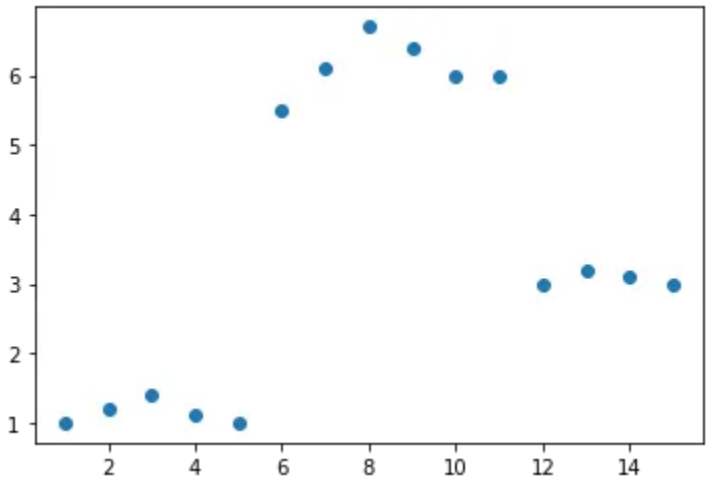


Công thức:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- Xây dựng cây

  Xem xét một tập dữ liệu trong đó chúng ta có 2 biến, như hình bên dưới:



  X là một biến liên tục, **Y** là một biến liên tục khác

X	Y
1	1
2	1.2
3	1.4
4	1.1
5	1
6	5.5
7	6.1
8	6.7
9	6.4
10	6
11	6
12	3
13	3.2
14	3.1

Bước 1:

Bước đầu tiên là sắp xếp dữ liệu dựa trên X (Trong trường hợp này, nó đã được sắp xếp sẵn). Sau đó, lấy giá trị trung bình của 2 hàng đầu tiên trong biến X (là (1+2)/2 = 1,5 theo tập dữ liệu đã cho). Chia tập dữ liệu thành 2 phần (Phần A và Phần B), cách nhau bởi $x < 1,5$ và $x \geq 1,5$.

Bây giờ, phần A chỉ bao gồm một điểm, đó là hàng đầu tiên (1,1) và tất cả các điểm khác nằm trong phần B. Bây giờ, lấy trung bình của tất cả các giá trị Y trong phần A và trung bình của tất cả các giá trị Y trong phần B một cách riêng biệt. 2 giá trị này lần lượt là kết quả dự đoán của cây quyết định với $x < 1,5$ và $x \geq 1,5$. Sử dụng các giá trị ban đầu và dự đoán, tính sai số bình phương trung bình và ghi lại.

Bước 2:

Ở bước 1, chúng ta đã tính giá trị trung bình cho 2 số đầu tiên của X được sắp xếp và phân chia tập dữ liệu dựa trên đó và tính toán các dự đoán. Sau đó thực hiện lại quy trình tương tự nhưng lần này, tính trung bình cho 2 số thứ hai của

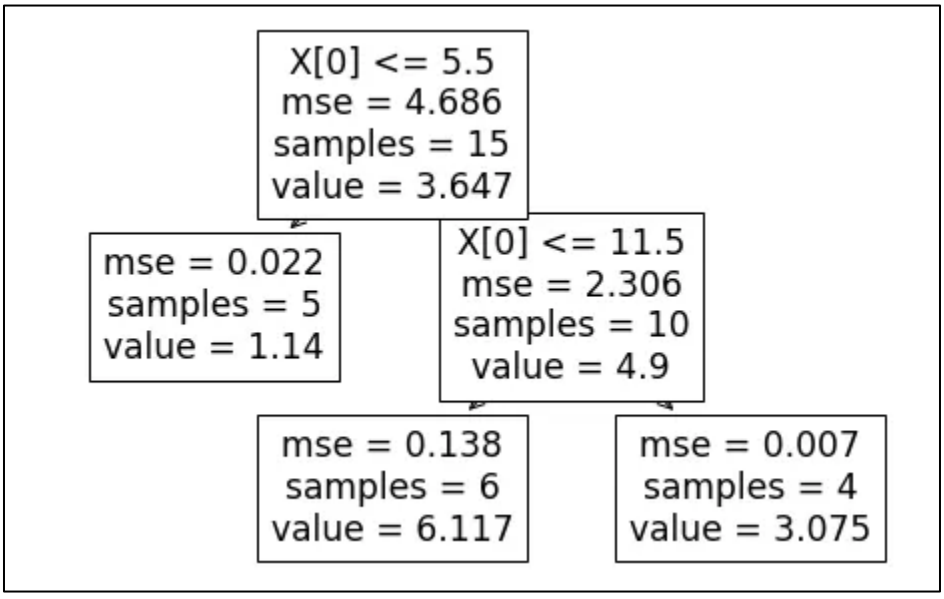
X đã sắp xếp ((2+3)/2 = 2,5). Sau đó, chia tập dữ liệu dựa trên x < 2,5 và x ≥ 2,5 thành Phần A và Phần B và dự đoán kết quả đầu ra, tìm lỗi bình phương trung bình như trong bước 1. Quá trình này được lặp lại cho 2 số thứ ba và thứ tư, thứ 4 và thứ 5 … cho đến cặp số thứ n - 1 và n.

Bước 3:

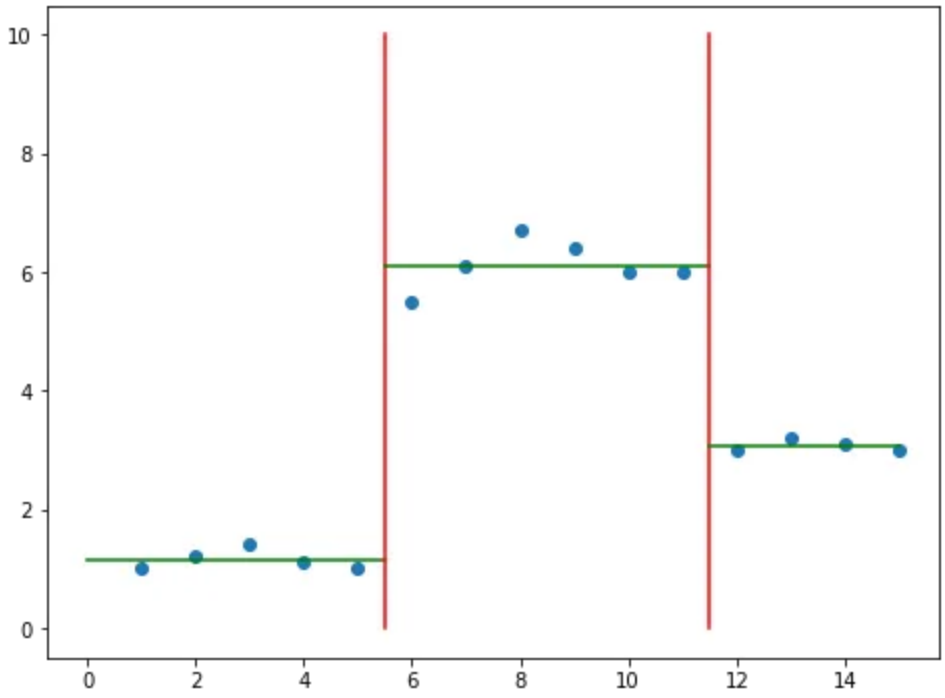
Bây giờ chúng ta đã tính được n -1 lỗi bình phương trung bình, chúng ta cần chọn điểm mà tại đó chúng ta sẽ chia tập dữ liệu. Và điểm đó là điểm dẫn đến sai số bình phương trung bình thấp nhất khi chia tại điểm đó. Trong trường hợp này, điểm là x = 5,5. Do đó cây sẽ được chia thành 2 phần. x < 5,5 và x ≥ 5,5. Nút Gốc được chọn theo cách này và các điểm dữ liệu đi về phía con trái và con phải của nút gốc được tiếp tục hiển thị đệ quy với cùng một thuật toán để phân tách thêm.

Bước 4: Xây dựng cây hồi quy

Cây hồi quy cho tập dữ liệu được hiển thị ở trên sẽ trông như sau



Và kết quả dự đoán sẽ như thế này



3.2.3 Ưu nhược điểm của Rừng ngẫu nhiên

Ưu điểm:

- Độ chính xác cao: Rừng ngẫu nhiên thường cho kết quả phân loại và dự đoán chính xác, đặc biệt khi áp dụng cho các tập dữ liệu lớn và phức tạp.
- Xử lý dữ liệu thiếu: Rừng ngẫu nhiên có khả năng xử lý dữ liệu thiếu hoặc giá trị ngoại lai một cách tốt. Thuật toán sử dụng phương pháp trung bình của các cây quyết định, giúp giảm ảnh hưởng của dữ liệu nhiễu lên kết quả cuối cùng.
- Kiểm tra mức độ quan trọng của biến: Rừng ngẫu nhiên cung cấp thông tin về mức độ quan trọng của các biến đầu vào. Điều này giúp hiểu rõ hơn về tác động của từng biến đến kết quả và đưa ra quyết định xây dựng mô hình.

Nhược điểm:

- Khó diễn giải: Rừng ngẫu nhiên không cung cấp một diễn giải rõ ràng và dễ hiểu về quá trình phân loại hoặc dự đoán. Điều này là do sự kết hợp của nhiều

cây quyết định và quyết định cuối cùng được dựa trên đa số phiếu bầu từ các cây con.

- Thời gian huấn luyện: Đối với tập dữ liệu lớn hoặc có số lượng biến đầu vào lớn, việc xây dựng rừng ngẫu nhiên có thể tốn nhiều thời gian. Điều này phụ thuộc vào số lượng cây quyết định và độ phức tạp của mô hình.
- Overfitting: Mặc dù Rừng ngẫu nhiên có khả năng giảm hiện tượng overfitting so với một cây quyết định đơn lẻ, nhưng nó vẫn có thể xảy ra khi số lượng cây quyết định quá lớn hoặc khi mô hình được điều chỉnh quá mức cho dữ liệu huấn luyện.

4. Tập dữ liệu

4.1 Thu thập dữ liệu

Sử dụng thư viện VnStock để thu thập dữ liệu về giá cổ phiếu từ ngày 01/01/2012 đến ngày 15/07/2023 của Tập đoàn Vin Group có mã trên sàn chứng khoán là VIC
.

Tập dữ liệu gồm 2827 hàng và 6 cột như sau:

- Giá mở cửa (Open)
- Giá cao nhất (High)
- Giá thấp nhất (Low)
- Giá đóng cửa (Close)
- Khối lượng giao dịch (Volume)
- Ngày giao dịch (Time)

```
[ ]  df = stock_historical_data("VIC", "2012-01-01", "2023-07-15")
     df = df.rename(columns={'time': 'Time'})
     df = df.rename(columns={'open': 'Open'})
     df = df.rename(columns={'high': 'High'})
     df = df.rename(columns={'low': 'Low'})
     df = df.rename(columns={'close': 'Close'})
     df = df.rename(columns={'volume': 'Volume'})
     df
```

	Time	Open	High	Low	Close	Volume
0	2012-03-20	16319	16985	16319	16652	130940
1	2012-03-21	16819	16819	16486	16652	144040
2	2012-03-22	16403	16819	16403	16652	123360
3	2012-03-23	16652	17152	16652	17152	153050
4	2012-03-26	17485	17818	16985	17818	219500
...	...	...	...	...	...	...
2822	2023-07-10	50500	51200	50200	50900	3345300
2823	2023-07-11	51000	51300	50800	50800	1927900
2824	2023-07-12	51100	51900	50800	51400	2295600
2825	2023-07-13	51500	52000	51200	51500	1894000
2826	2023-07-14	51800	51800	51000	51400	2013700

2827 rows × 6 columns

Figure 1: tập dữ liệu

4.2 Tiền xử lí dữ liệu

Dữ liệu được thu thập có thông tin như sau:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2827 entries, 0 to 2826
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Time    2827 non-null   object
 1   Open    2827 non-null   int64 
 2   High    2827 non-null   int64 
 3   Low     2827 non-null   int64 
 4   Close   2827 non-null   int64 
 5   Volume  2827 non-null   int64 
dtypes: int64(5), object(1)
memory usage: 132.6+ KB
```

Figure 2: Thông tin tập dữ liệu

Tập dữ liệu gồm 2827 hàng, 6 cột không có ô nào có giá trị NULL. Các cột Open, High, Low, Close, Volume có kiểu dữ liệu số, tuy nhiên Time có kiểu dữ liệu phi số .

Thực hiện phân tích cột Time thành 3 cột Year, Month, Day với kiểu dữ liệu số.

```
[8] import pandas as pd

    df['Time'] = pd.to_datetime(df['Time'], format='%Y-%m-%d')

    df['Day'] = df['Time'].apply(lambda x: x.day)
    df['Month'] = df['Time'].apply(lambda x: x.month)
    df['Year'] = df['Time'].apply(lambda x: x.year)
```

```
[ ] df
```

	Time	Open	High	Low	Close	Volume	Day	Month	Year
0	2012-03-20	16319	16985	16319	16652	130940	20	3	2012
1	2012-03-21	16819	16819	16486	16652	144040	21	3	2012
2	2012-03-22	16403	16819	16403	16652	123360	22	3	2012
3	2012-03-23	16652	17152	16652	17152	153050	23	3	2012
4	2012-03-26	17485	17818	16985	17818	219500	26	3	2012
...	...	...	...	...	...	...	...	...	...
2822	2023-07-10	50500	51200	50200	50900	3345300	10	7	2023
2823	2023-07-11	51000	51300	50800	50800	1927900	11	7	2023
2824	2023-07-12	51100	51900	50800	51400	2295600	12	7	2023
2825	2023-07-13	51500	52000	51200	51500	1894000	13	7	2023
2826	2023-07-14	51800	51800	51000	51400	2013700	14	7	2023

2827 rows × 9 columns

4.3 Trực quan hóa dữ liệu

Biểu đồ biến động giá cổ phiếu VIC theo thời gian

```
plt.figure(figsize=(12,8))
plt.plot(df['Close'])
plt.xlabel('Year')
plt.ylabel('Price ($)')
plt.rcParams.update({'font.size': 14})
```

Figure 3: biểu đồ biến động giá cổ phiếu VIC theo thời gian

Vẽ biểu đồ scatter spot biểu diễn sự phụ thuộc của Giá đóng cửa (Close) vào các đặc trưng còn lại:

```
[ ]  fig = plt.figure(figsize=(35, 25))
     sns.set(style='whitegrid')

     num_cols = len(df.columns) - 1
     num_rows = (num_cols + 3) // 4

     for i, col in enumerate(df.columns[:-1]):
         fig.add_subplot(num_rows, 4, i + 1)
         plt.scatter(df[col], df['Close'])
         plt.xlabel(col)
         plt.ylabel('close')

     plt.tight_layout()
```

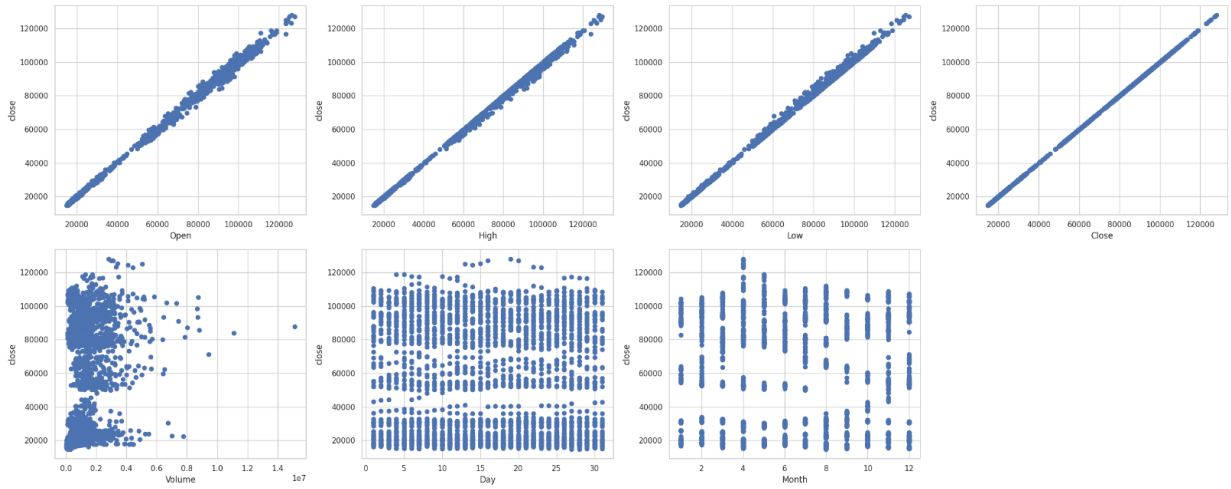


Figure 4: Biểu đồ Scatter Plot biểu diễn sự phụ thuộc của giá đóng cửa cổ phiếu vào các đặc trưng còn lại

Có thể thấy giá đóng cửa (Close) phụ thuộc tuyến tính vào các đặc trưng Open, High, Low.

5. Phương pháp

Giá đóng cửa của cổ phiếu phụ thuộc vào nhiều yếu tố, đặc biệt những năm gần đây, do ảnh hưởng của dịch covid, giá cổ phiếu có nhiều biến động. Cụ thể trước năm 2021 dịch bình thường tuy nhiên đến năm 2022 dịch bùng mạnh dẫn đến giá cổ phiếu giảm.

Sử dụng 6 đặc trưng sau để dự đoán giá đóng cửa của cổ phiếu:

- Open : Giá mở cửa
- High : Giá cao nhất
- Low : Giá thấp nhất
- Volume: Khối lượng giao dịch
- Year : Năm giao dịch
- Month : Tháng giao dịch
- Day : Ngày giao dịch

Tạo tập X (dữ liệu đã biết - đầu vào) bằng cách loại bỏ cột Close khỏi dataframe, tập Y (dữ liệu cần dự đoán – đầu ra) là cột Close (giá đóng cửa):

```
[15] x_data = df.drop('Close', axis=1, inplace=False)
     x_data
```

Time	Open	High	Low	Volume	Day	Month	Year
2012-03-20	16319	16985	16319	130940	20	3	2012
2012-03-21	16819	16819	16486	144040	21	3	2012
2012-03-22	16403	16819	16403	123360	22	3	2012
2012-03-23	16652	17152	16652	153050	23	3	2012
2012-03-26	17485	17818	16985	219500	26	3	2012
...	...	...	...	...	...	...	...
2023-07-10	50500	51200	50200	3345300	10	7	2023
2023-07-11	51000	51300	50800	1927900	11	7	2023
2023-07-12	51100	51900	50800	2295600	12	7	2023
2023-07-13	51500	52000	51200	1894000	13	7	2023
2023-07-14	51800	51800	51000	2013700	14	7	2023

2827 rows × 7 columns

```
data=df.filter(['Close'])
y_data = data.copy()
y_data
```

Time	Close
2012-03-20	16652
2012-03-21	16652
2012-03-22	16652
2012-03-23	17152
2012-03-26	17818
...	...
2023-07-10	50900
2023-07-11	50800
2023-07-12	51400
2023-07-13	51500
2023-07-14	51400

2827 rows × 1 columns

Chia các cặp dữ liệu (X,Y) thành hai tập với tỷ lệ:

- 80% cho tập train: sử dụng trong huấn luyện mô hình
- 20% cho tập test: sử dụng trong đánh giá mô hình

```
from sklearn.model_selection import train_test_split

x_data_train, x_data_test, y_data_train, y_data_test = train_test_split(x_data, y_data, test_size=0.2, shuffle=False)
```

Figure 5: Chia tập huấn luyện và đánh giá

5.1 Xây dựng mô hình Hồi quy tuyến tính

Dựa vào biểu đồ Scatter Plot biểu diễn sự phụ thuộc của giá đóng cửa cổ phiếu vào các đặc trưng còn lại. Có thể thấy sự phụ thuộc tuyến tính của giá đóng cửa (Close) vào 3 đặc trưng:

- Open: Giá mở cửa

- High: Giá cao nhất
- Low: Giá thấp nhất

Vì vậy thực hiện loại bỏ các cột Year, Month, Day khỏi tập x_data_train và x_data_test cho mô hình hồi quy tuyến tính:

```
[47] x_data_train_1 = x_data_train.drop(['Year', 'Month', 'Day', 'Volume'], axis=1, inplace=False)
```

```
[48] x_data_test_1 = x_data_test.drop(['Year', 'Month', 'Day', 'Volume'], axis=1, inplace=False)
```

```
[49] x_data_train_1
```

Time	Open	High	Low
2012-03-20	16319	16985	16319
2012-03-21	16819	16819	16486
2012-03-22	16403	16819	16403
2012-03-23	16652	17152	16652
2012-03-26	17485	17818	16985
...	...	...	...
2021-04-05	111987	112165	109498
2021-04-06	110476	115098	110209
2021-04-07	112876	113764	111720
2021-04-08	114120	114120	110209
2021-04-09	110920	111454	109409

2261 rows × 3 columns

Thực hiện huấn luyện mô hình Hồi quy tuyến tính dựa đoán giá đóng cửa (Close) dựa vào quan sát 3 đặc trưng: Open, High, Low.

```
[51] reg = LinearRegression()
     reg.fit(x_data_train_1,y_data_train)
```

```
▾ LinearRegression
LinearRegression()
```

Figure 6: Xây dựng mô hình Hồi quy tuyến tính dự đoán giá đóng cửa

5.2 Xây dựng mô hình Rừng ngẫu nhiên

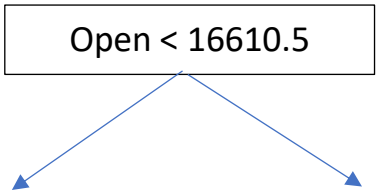
Ví dụ chọn tham số để xây dựng 1 cây quyết định:

	Open	Close
8	15986	15986
9	15986	15986
6	16153	16153
0	16319	16652
2	16403	16652
7	16569	15736
3	16652	17152
1	16819	16652
4	17485	17818
5	17818	16985

⇨ **Sum square error tương ứng:**

```
MAE: 410.733
MAE: 356
MAE: 329.238
MAE: 383.05
MAE: 416.32
MAE: 316.433
MAE: 425.867
MAE: 408
MAE: 453.311
```

⇨ **Giá trị thuộc tính Open được chọn phân tách là trung bình của hàng 7 và 3**



Tạo một đối tượng regressor của RandomForestRegressor với các tham số sau:

- n_estimators=50: số lượng cây quyết định được tạo ra trong mô hình Random Forest. Giá trị này xác định số lượng cây quyết định độc lập trong mô hình. Đối với một giá trị nhất định, việc tăng số lượng cây có thể cải thiện độ chính xác của mô hình, nhưng cũng đồng nghĩa với việc tăng thời gian huấn luyện.
- random_state=42: giá trị random_state cố định, đảm bảo rằng mô hình sẽ tạo ra kết quả nhất quán mỗi khi chạy.

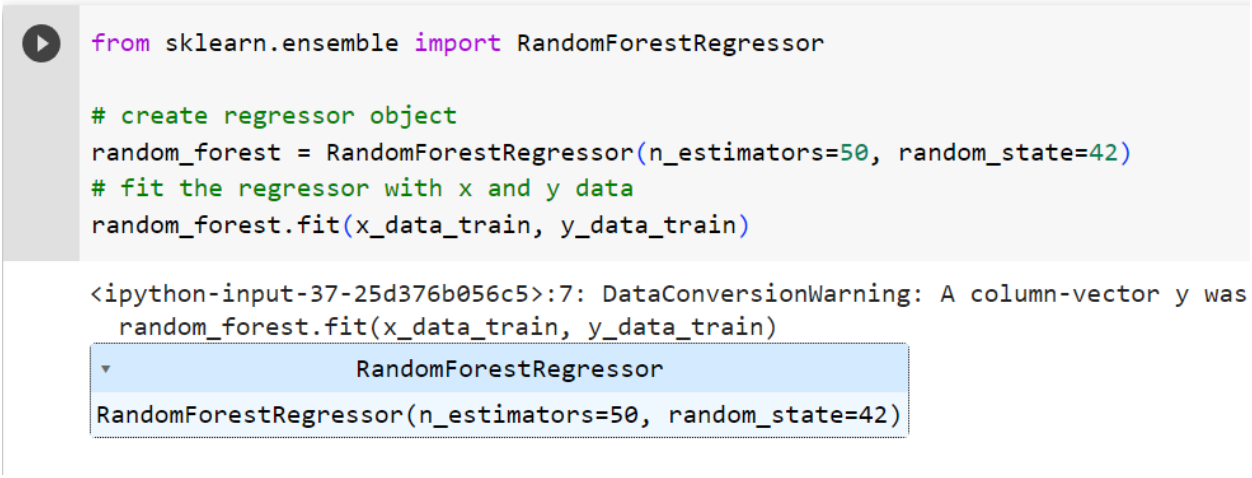


Figure 7: Xây dựng mô hình Rừng ngẫu nhiên dự đoán giá đóng cửa

6. Đánh giá mô hình

6.1 Hồi quy tuyến tính

Sử dụng giá trị MAE (Mean Absolute Error) đo lường sai số trung bình giữa giá trị dự đoán và giá trị thực tế. Nó được tính bằng cách lấy tổng trị tuyệt đối của hiệu giữa giá trị dự đoán và giá trị thực tế, sau đó chia cho số lượng mẫu. Kết quả đánh giá giá trị MAE trên hai tập train và tập test như sau:

- MAE trên tập train = 320.02
- MAE trên tập test = 618.26

Sử dụng giá trị R^2(R-squared) đo lường tỷ lệ phương sai của biến phụ thuộc mà mô hình có thể dự đoán. Nó được tính bằng cách so sánh sai số trung bình giữa mô hình và một mô hình đơn giản nhất (mô hình mà chỉ dự đoán giá trị trung bình của biến phụ thuộc).

- R^2 trên tập train : 0.999
- R^2 trên tập test: 0.998

```
[57] print('MAE in training set: ',mean_absolute_error(y_train_pre,y_data_train))
     print('MAE in test set: ',mean_absolute_error(y_test_pre,y_data_test))

     MAE in training set:  320.0203989278584
     MAE in test set:  618.266887507804

[58] from sklearn.metrics import r2_score
     print('R2_score in training set: ',r2_score(y_train_pre,y_data_train))
     print('R2_score in test set: ',r2_score(y_test_pre,y_data_test))

     R2_score in training set:  0.9997621285527565
     R2_score in test set:  0.9983375904650929
```

Figure 8: Đánh giá kết quả mô hình Hồi quy tuyến tính

Vẽ biểu đồ so sánh độ chính xác của việc dự đoán giá đóng cửa dựa trên tập Test:

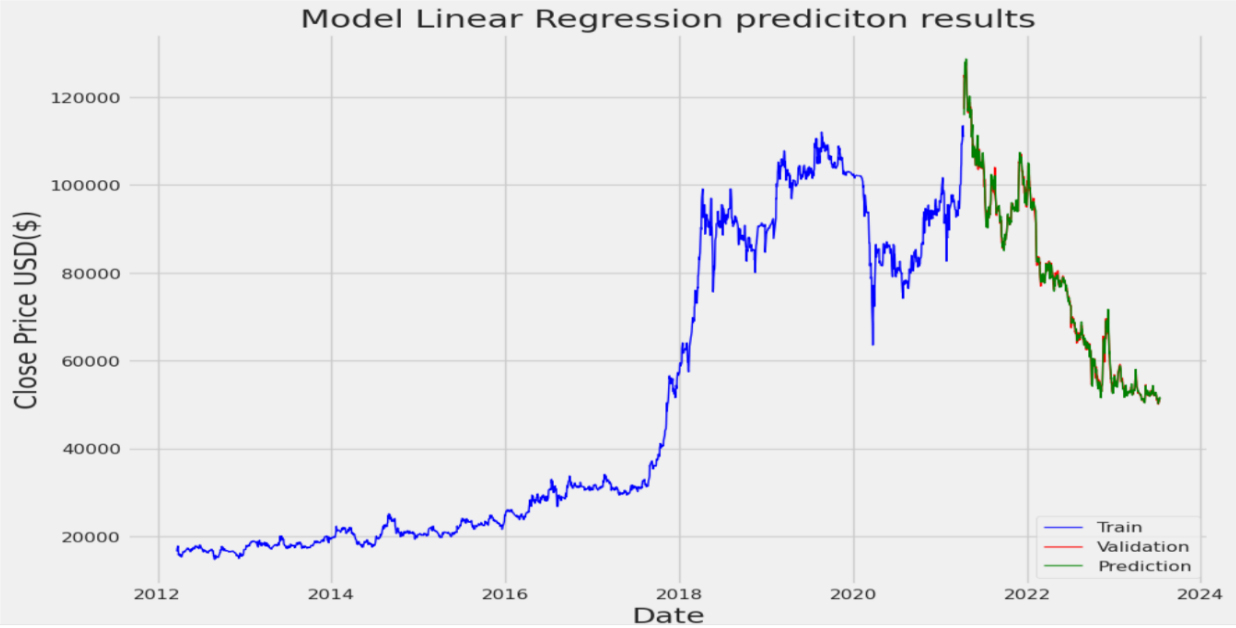


Figure 9: Biểu đồ giá đóng cửa theo thời gian so sánh kết quả dự đoán theo mô hình Hồi quy tuyến tính và tập test

6.2 Rừng ngẫu nhiên

- R^2 trên tập train : 0.999
- R^2 trên tập test: 0.987

```
from sklearn.metrics import r2_score
print('R2_score in training set: ',r2_score(y_pre_train_rd,y_data_train))
print('R2_score in test set: ',r2_score(y_pre_test_rd,y_data_test))
```

```
R2_score in training set:  0.9999539673775647
R2_score in test set:  0.9873870477775423
```

```
[23] from sklearn.metrics import r2_score
     print('R2_score in training set: ',r2_score(y_pre_train_rd,y_train))
     print('R2_score in test set: ',r2_score(y_pre_test_rd,y_test))
```

```
R2_score in training set:  0.9997662964130263
R2_score in test set:  0.9022959666033611
```

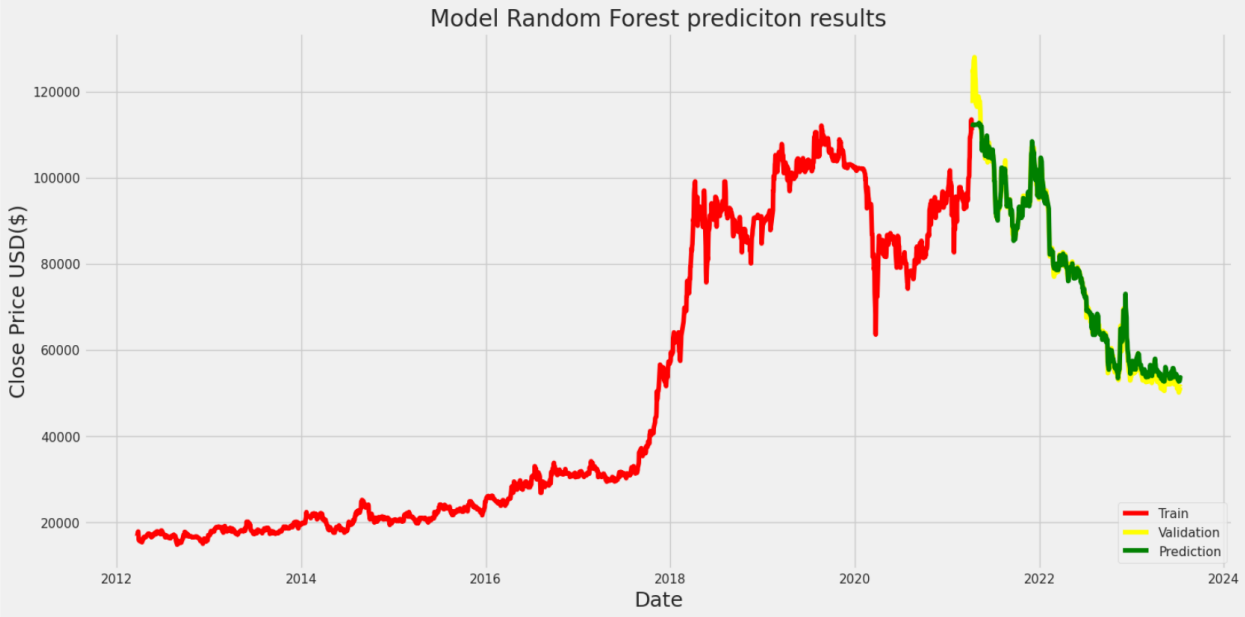


Figure 10: Biểu đồ giá đóng cửa theo thời gian so sánh kết quả dự đoán theo mô hình Rừng ngẫu nhiên và tập test

7. Ứng dụng và kết luận

Sử dụng hai mô hình vừa huấn luyện để dự đoán giá cổ phiếu cho ngày hôm sau. Tuy nhiên việc dự đoán này chỉ là một nguồn tham khảo, không thật sự đáng tin cậy dù độ chính xác trên tập test là khá cao vì giá cổ phiếu còn phụ thuộc rất nhiều vào tình hình kinh tế - chính trị xã hội, các yếu tố khách quan khác.

```
[ ]  last_day_features = x_data[-1:]

     predicted_price = reg.predict(last_day_features)

     print("Predicted price for the next day:", predicted_price)

     Predicted price for the next day: [[51191.4619389]]
```

Figure 11: sử dụng mô hình hồi quy tuyến tính dự đoán giá cổ phiếu cho ngày hôm sau

```
last_day_features = x_data[-1:]

predicted_price = random_forest.predict(last_day_features)

print("Predicted price for the next day:", predicted_price)

Predicted price for the next day: [53434.3]
```

Figure 12: sử dụng mô hình rừng ngẫu nhiên dự đoán giá cổ phiếu cho ngày hôm sau

Các mô hình dự đoán Hồi quy tuyến tính và Rừng ngẫu nhiên đã thực hiện có độ chính xác khá cao, tuy nhiên việc dự đoán giá cổ phiếu là một bài toán phức tạp bởi cần kết hợp các phân tích khác, nắm vững thông tin thị trường, kết hợp các công cụ và kỹ thuật dự đoán khác nhau.

Lời cảm ơn

Chúng em xin gửi lời cảm ơn thầy Nguyễn Bá Ngọc về những góp ý, hướng dẫn, định hướng giúp chúng em thực hiện bài toán trong quá trình thực hiện Project 2. Từ đó, chúng em có thể áp dụng những kiến thức đã học vào bài toán thực tế.

Do quá trình tìm hiểu kiến thức còn nhiều hạn chế, bài báo cáo của chúng em còn nhiều thiếu sót, mong thầy nhận xét để bài báo cáo được hoàn thiện hơn.