**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Project I**

**Chương trình dự đoán kết quả xổ số**

**Hoàng Anh Quyền**

[quyen.ha224893@sis.hust.edu.vn](mailto:quyen.ha224893@sis.hust.edu.vn)

MSSV: 20224875

**Ngành Khoa học máy tính**

**Trường Công nghệ Thông tin và Truyền thông**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Nguyễn Phi Lê |
| **Khoa:** | Khoa học máy tính |
| **Trường:** | Trường Công nghệ Thông tin  và Truyền thông |
| **HÀ NỘI, 12/2024**  Mục lục  [CHƯƠNG 1. Thu thập dữ liệu từ trang web xổ số 3](#_Toc186116182)  [1.1 Xác định trang web và phần dữ liệu cần thu thập 3](#_Toc186116183)  [1.1.1 Trang web xổ số 3](#_Toc186116184)  [1.1.2 Phần dữ liệu cần thu thập 3](#_Toc186116185)  [1.2 Tiến hành thu thập dữ liệu 4](#_Toc186116186)  [1.2.1 Khai báo các thư viện cần thiết 4](#_Toc186116187)  [1.2.2 Truy cập vào trang web bằng Selenium 4](#_Toc186116188)  [1.2.3 Thu thập kết quả xổ số trong 5000 ngày qua 5](#_Toc186116189)  [CHƯƠNG 2. Huấn luyện Model và dự đoán kết quả xổ số dựa trên dữ liệu đã có 8](#_Toc186116190)  [2.1 Lý thuyết về phương pháp áp dụng để train model 8](#_Toc186116191)  [2.1.1 Linear Regression (Hồi quy tuyến tính) 8](#_Toc186116192)  [2.1.2 Polynomial Regression (Hồi quy đa thức) 10](#_Toc186116193)  [2.1.3 Decision Tree ( Cây quyết định) 11](#_Toc186116194)  [2.1.4 Random Forest 12](#_Toc186116195)  [2.2 Cài đặt 14](#_Toc186116196)  [2.2.1 Khai báo các thư viện cần thiết: 14](#_Toc186116197)  [2.2.2 Nhập dữ liệu bằng thư viện pandas 15](#_Toc186116198)  [2.2.3 Phân chia dữ liệu 15](#_Toc186116199)  [2.2.4 Huấn luyện các mô hình 16](#_Toc186116200)  [2.2.5 Sử dụng các mô hình để dự đoán kết quả xổ số hôm nay 17](#_Toc186116201)  [2.2.6 Đánh giá chất lượng các mô hình 19](#_Toc186116202)  [CHƯƠNG 3. Kết Luận 19](#_Toc186116203) | |
|  | |

# Thu thập dữ liệu từ trang web xổ số

## Xác định trang web và phần dữ liệu cần thu thập

### Trang web xổ số

Trang web: https://xoso.com.vn/xsmb-5000-ngay.html

Đây là giao diện của trang web

A screenshot of a computer

Description automatically generated

### Phần dữ liệu cần thu thập

Mỗi ngày thì sẽ có một bảng tương ứng với kết quả xổ số của ngày đó

A screenshot of a computer

Description automatically generated

## Tiến hành thu thập dữ liệu

### Khai báo các thư viện cần thiết

A black background with orange and white text

Description automatically generated

* **Selenium** là một thư viện phổ biến trong Python được sử dụng để tự động hóa việc duyệt web. Nó cho phép bạn điều khiển trình duyệt web (Chrome, Firefox, v.v.) thông qua mã Python, thực hiện các tác vụ như điều hướng trang web, tương tác với các phần tử trên trang (như click, nhập văn bản), và kiểm tra các ứng dụng web.
* **Beautiful Soup** là một thư viện Python mạnh mẽ được sử dụng để phân tích cú pháp (parsing) HTML và XML. Nó rất hữu ích khi bạn cần trích xuất dữ liệu từ các trang web (web scraping), xử lý các tài liệu HTML và XML, và duyệt qua các phần tử của trang web một cách dễ dàng.
* Thư viện **time** trong Python cung cấp các chức năng liên quan đến thời gian, giúp bạn thực hiện các thao tác như đo lường thời gian, trì hoãn (sleep), hoặc làm việc với các đối tượng thời gian. Đây là một thư viện rất hữu ích khi bạn cần làm việc với thời gian hệ thống hoặc cần đo lường hiệu suất của chương trình.

### Truy cập vào trang web bằng Selenium

A screen shot of a computer

Description automatically generated

Đoạn code này tạo một đối tượng chrome\_options để cấu hình trình duyệt Chrome.Cấu hình đặc biệt là giữ lại cửa sổ trình duyệt mở sau khi mã thực thi xong (thông qua detach=True).Sau đó, Selenium mở trang web xổ số miền Bắc (https://xoso.com.vn/xsmb-5000-ngay.html) trong trình duyệt Chrome. Cuối cùng, trình duyệt sẽ không tự động đóng mà vẫn mở cho đến khi bạn đóng nó bằng tay.

### Thu thập kết quả xổ số trong 5000 ngày qua

Sau khi truy cập vào trang web thì chúng ta cần click vào nút “Xem thêm” để hiện thị toàn bộ kết quả của 5000 ngày qua.

A screenshot of a computer

Description automatically generated

Để làm được điều này chúng ta sẽ sử dụng thư viện selenium và thư viện time để thực hiện:

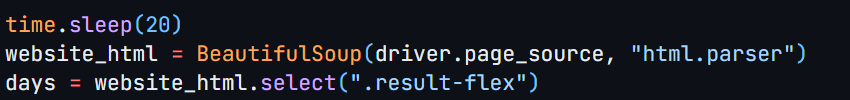
A black background with text

Description automatically generated

Ở đây, trước tiên chúng ta sẽ thử tìm kiếm vị trí của nút “Xem thêm” bằng XPATH nếu nó đã tồn tại thì sẽ click vào, còn nếu nó chưa xuất hiện (trường hợp trang web chưa load được toàn bộ thông tin) thì sẽ chờ 2 giây và thực hiện lại.

A screenshot of a computer

Description automatically generated



Tạm dừng trong 20 giây để tải toàn bộ thông tin. Sau khi hiện thị hết toàn bộ dữ liệu thì sẽ tải mã nguồn HTML của trang web hiện tại. Sử dụng BeautifulSoup để phân tích cú pháp HTML. Tìm kiếm và lưu trữ các phần tử có lớp CSS .result-flex vào biến days. Tiếp theo sẽ phân tích, định dạng, xử lý dữ liệu đưa về kiểu mong muốn

A screen shot of a computer code

Description automatically generated

* Duyệt qua các dòng dữ liệu (days), thu thập thông tin từ các ô bảng.
* Xử lý văn bản thu thập được, loại bỏ khoảng trắng và định dạng lại một số phần (như việc lặp lại một chuỗi con theo yêu cầu).
* Lấy thông tin ngày tháng từ các phần tử liên quan.
* Nối tất cả các kết quả lại và lưu vào biến all\_result, với mỗi dòng dữ liệu được cách nhau bởi một dấu xuống dòng.

A computer screen shot of code

Description automatically generated

Hàm format là một phần quan trọng trong việc xử lý và định dạng lại các chuỗi dữ liệu, đặc biệt là với các chuỗi có yêu cầu lặp lại thông tin (dựa trên số lần lặp trong ngoặc).

A computer code on a black background

Description automatically generated

Sau khi đã xử lý hết toàn bộ dữ liệu thì sẽ được đẩy vào file data.csv. Và đây là kết quả:

A screenshot of a computer screen

Description automatically generated

# Huấn luyện Model và dự đoán kết quả xổ số dựa trên dữ liệu đã có

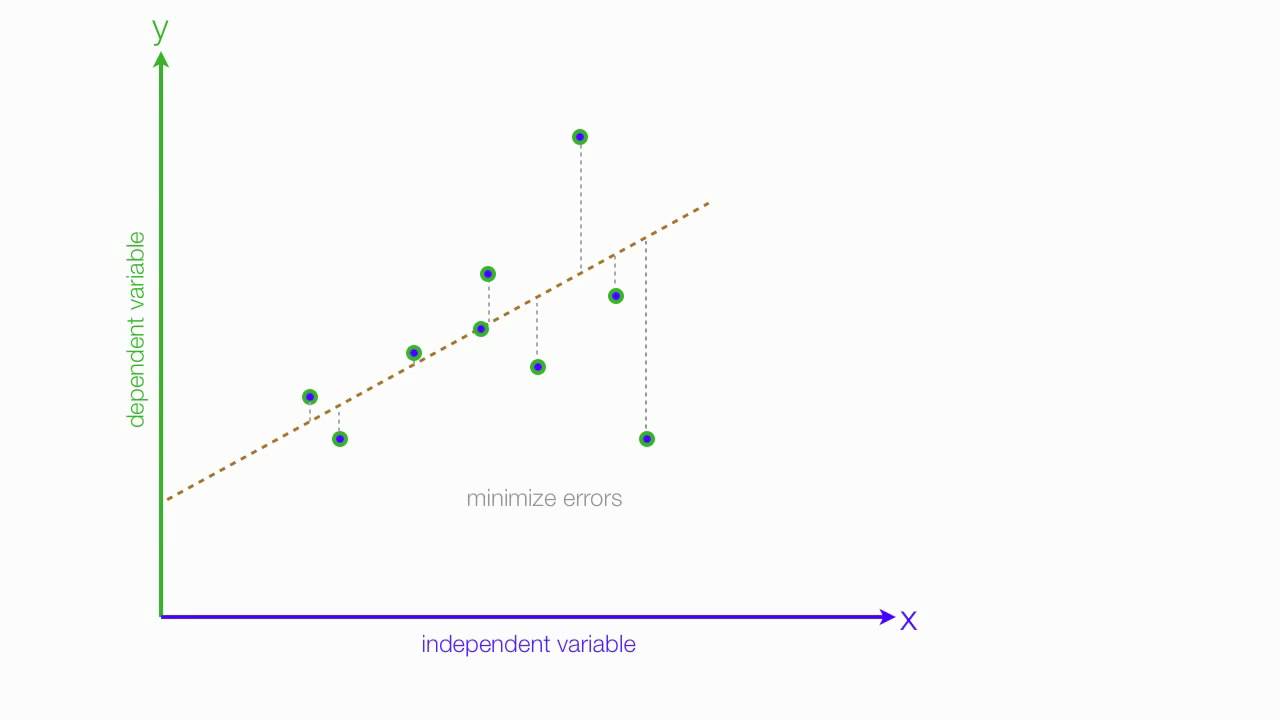
## Lý thuyết về phương pháp áp dụng để train model

### Linear Regression (Hồi quy tuyến tính)

#### Khái niệm

**Linear Regression** (hồi quy tuyến tính) là một kỹ thuật trong thống kê và học máy được sử dụng để mô hình hóa mối quan hệ giữa một hoặc nhiều biến độc lập (predictors, features) và một biến phụ thuộc (target, output). Hồi quy tuyến tính tìm cách xây dựng một phương trình đường thẳng tối ưu dựa trên dữ liệu, giúp dự đoán giá trị của biến phụ thuộc.

#### Minh họa



#### Phương trình hồi quy tuyến tính

Đối với trường hợp đơn giản với một biến độc lập (hồi quy tuyến tính đơn), phương trình có dạng:

y=β0​+β1​x+ϵ

* y: Biến phụ thuộc (target variable).
* x: Biến độc lập (predictor).
* β0​: Hệ số chặn (intercept) – giá trị của y khi x=0.
* β1: Hệ số góc (slope) – sự thay đổi của y khi x thay đổi 1 đơn vị.
* ϵ: Sai số ngẫu nhiên (error term).

Đối với hồi quy tuyến tính đa biến (với nhiều biến độc lập), phương trình sẽ mở rộng thành:

y=β0​+β1​x1​+β2​x2​+⋯+βn​xn​+ϵ

* x1,x2,…,xn​ là các biến độc lập (predictors).
* β1,β2,…,βn​ là các hệ số hồi quy tương ứng.

#### Cách thức hoạt động

Hồi quy tuyến tính tìm kiếm các hệ số β0,β1,…,βn sao cho phương trình mô hình hóa được mối quan hệ giữa các biến và có thể dự đoán chính xác giá trị của y từ x. Để làm được điều này, hồi quy tuyến tính sử dụng một phương pháp tối ưu, thường là **phương pháp tối thiểu bình phương** (Ordinary Least Squares - OLS), để giảm thiểu tổng bình phương sai số giữa giá trị dự đoán và giá trị thực tế.

#### Ứng dụng của hồi quy tuyến tính

* **Dự đoán**: Dự đoán giá trị của biến phụ thuộc dựa trên các biến độc lập. Ví dụ, dự đoán giá trị nhà dựa trên diện tích, số phòng, vị trí, v.v.
* **Mô hình hóa mối quan hệ**: Hiểu rõ mối quan hệ giữa các biến. Ví dụ, trong nghiên cứu khoa học, có thể dùng hồi quy tuyến tính để hiểu tác động của một yếu tố (như nhiệt độ) đối với sự thay đổi của một yếu tố khác (như sản lượng nông sản).
* **Phân tích sự ảnh hưởng**: Kiểm tra xem các yếu tố độc lập có ảnh hưởng đáng kể đến biến phụ thuộc hay không. Đặc biệt là qua các hệ số β, có thể xác định yếu tố nào có ảnh hưởng lớn nhất.

#### Ưu điểm của hồi quy tuyến tính

* **Đơn giản và dễ hiểu**: Mô hình tuyến tính dễ dàng xây dựng và giải thích.
* **Tính toán nhanh**: Vì đây là mô hình đơn giản, nên tính toán được thực hiện nhanh chóng.
* **Ứng dụng rộng rãi**: Hồi quy tuyến tính có thể áp dụng trong nhiều tình huống và bài toán thực tế.

#### Nhược điểm của hồi quy tuyến tính

* **Giả định tuyến tính**: Hồi quy tuyến tính giả định rằng mối quan hệ giữa các biến là tuyến tính. Nếu mối quan hệ là phi tuyến, mô hình sẽ không chính xác.
* **Nhạy cảm với ngoại lệ (outliers)**: Mô hình hồi quy tuyến tính có thể bị ảnh hưởng lớn bởi các giá trị ngoại lệ.
* **Hội tụ đa cộng tuyến (multicollinearity)**: Nếu có sự tương quan mạnh giữa các biến độc lập, mô hình có thể gặp khó khăn trong việc xác định các hệ số hồi quy chính xác.

#### Ví dụ về hồi quy tuyến tính

Giả sử bạn muốn dự đoán giá của một ngôi nhà (y) dựa trên diện tích của nó (x). Sau khi thu thập dữ liệu, bạn có thể áp dụng hồi quy tuyến tính để tìm một phương trình như sau:

y=50,000+1,200xy = 50,000 + 1,200xy=50,000+1,200x

Ở đây:

* 50,000 là giá trị căn bản của ngôi nhà khi diện tích là 0.
* 1,200 là mức thay đổi giá mỗi khi diện tích tăng lên 1 mét vuông.

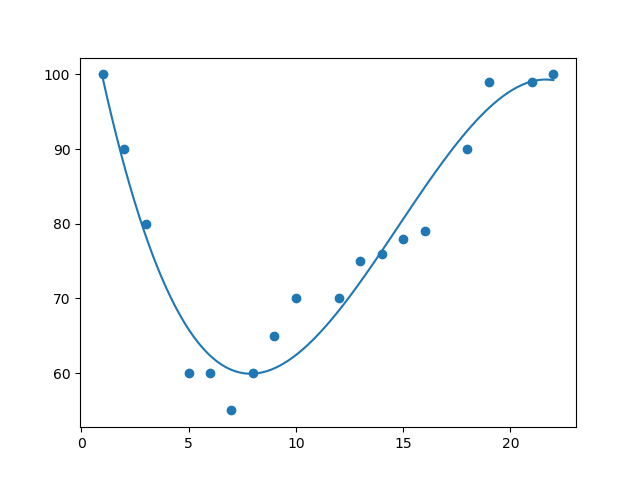
Khi bạn có diện tích của ngôi nhà, bạn có thể sử dụng phương trình này để dự đoán giá trị của nó.

### Polynomial Regression (Hồi quy đa thức)

#### 2.1.2.1 Khái niệm

**Polynomial regression** (hồi quy đa thức) là một phương pháp hồi quy trong đó mối quan hệ giữa biến độc lập (hoặc các biến độc lập) và biến phụ thuộc được mô hình hóa bằng một đa thức bậc n. Phương pháp này được sử dụng khi dữ liệu có mối quan hệ không phải là đường thẳng mà là một đường cong.

#### Minh họa



#### Công thức hồi quy đa thức

Trong hồi quy đa thức, mô hình được mở rộng với các bậc cao hơn của biến x, ví dụ như sau (cho bậc 2):

y=β0​+β1​x+β2​x2

Với mô hình này, x2 là bậc 2 của biến độc lập x. Bạn có thể mở rộng mô hình lên các bậc cao hơn, ví dụ:

y=β0​+β1​x+β2​x2+β3​x3+⋯+βn​xn

#### Các bước trong hồi quy đa thức

1. **Chọn bậc của đa thức (n)**: Bậc n của đa thức xác định số lượng các thành phần trong mô hình, ví dụ như x2, x3 v.v. Bậc càng cao thì mô hình càng linh hoạt, nhưng cũng dễ dẫn đến hiện tượng **quá khớp** (overfitting).
2. **Biến đổi đặc trưng**: Hồi quy đa thức hoạt động bằng cách biến đổi các đặc trưng độc lập thành các bậc cao hơn (ví dụ: x2 , x3) . Với một đặc trưng duy nhất x, ta sẽ chuyển nó thành:

Xpoly=[x,x2,x3,…,xn]

1. **Huấn luyện mô hình hồi quy**: Sử dụng các thuật toán hồi quy (ví dụ: bình phương tối thiểu hoặc gradient descent) để tìm các hệ số β0​,β1​,…,βn​.
2. **Dự đoán**: Sau khi huấn luyện mô hình, bạn có thể sử dụng mô hình để dự đoán giá trị cho các điểm dữ liệu mới.

#### Ưu điểm của hồi quy đa thức

* **Phù hợp với mối quan hệ phi tuyến**: Hồi quy đa thức có thể mô hình hóa những mối quan hệ phi tuyến mà hồi quy tuyến tính không làm được.
* **Linh hoạt**: Có thể sử dụng cho nhiều loại dữ liệu phi tuyến khác nhau.

#### 2.1.2.5 Nhược điểm của hồi quy đa thức

* **Quá khớp (Overfitting)**: Nếu bậc đa thức quá cao, mô hình có thể dễ dàng quá khớp với dữ liệu huấn luyện, tức là mô hình sẽ học cả nhiễu (noise) thay vì chỉ học mối quan hệ thực sự.
* **Khó giải thích**: Các mô hình đa thức với bậc cao có thể trở nên phức tạp và khó hiểu.

### Decision Tree ( Cây quyết định)

#### Khái niệm

**Cây quyết định** là một mô hình dạng cây được sử dụng trong phân tích, dự đoán, và ra quyết định. Nó là một công cụ trực quan và dễ hiểu, được ứng dụng rộng rãi trong học máy (machine learning), khai phá dữ liệu (data mining), và thống kê.

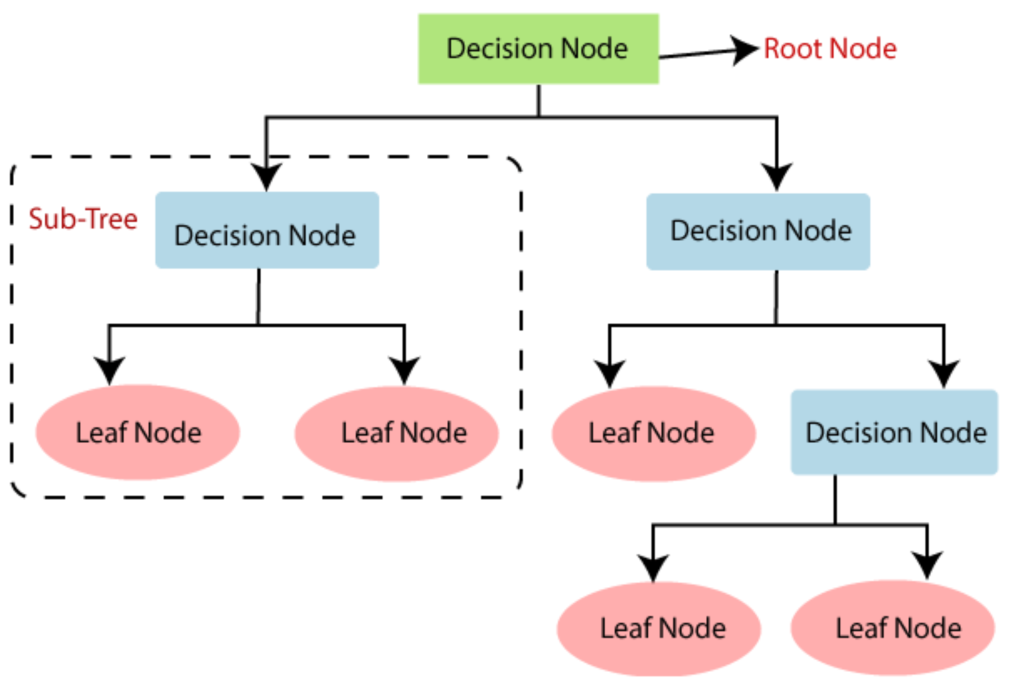
**Cấu trúc của một cây quyết định**

1. **Nút gốc (Root Node):** Điểm xuất phát của cây, chứa toàn bộ tập dữ liệu và đại diện cho một câu hỏi hoặc điều kiện cần phân nhánh.
2. **Nút quyết định (Decision Nodes):** Các nút trung gian, mỗi nút đại diện cho một câu hỏi hoặc điều kiện dựa trên một thuộc tính cụ thể.
3. **Nhánh (Branches):** Các đường nối từ một nút đến các nút con, biểu thị các kết quả của điều kiện hoặc câu hỏi.
4. **Nút lá (Leaf Nodes):** Các nút cuối cùng, đại diện cho kết quả hoặc quyết định cuối cùng (như phân loại hoặc giá trị dự đoán).

#### 2.1.3.2 Cách hoạt động:

* Một cây quyết định bắt đầu tại nút gốc.
* Ở mỗi nút quyết định, nó kiểm tra một thuộc tính và phân nhánh dựa trên giá trị của thuộc tính đó.
* Quá trình tiếp tục cho đến khi đạt đến nút lá, nơi cây đưa ra dự đoán hoặc kết luận.

#### Minh họa về cây quyết định



#### Ví dụ thực tế

* **Trong kinh doanh:** Quyết định có nên cho khách hàng vay tiền dựa trên lịch sử tín dụng, thu nhập, và nợ nần của họ.
* **Trong y tế:** Chẩn đoán bệnh dựa trên các triệu chứng của bệnh nhân.
* **Trong giáo dục:** Xác định khả năng học sinh đạt điểm cao dựa trên giờ học và sự tham gia lớp học.

#### Ưu điểm

* Trực quan, dễ hiểu.
* Không yêu cầu xử lý dữ liệu quá phức tạp.
* Có thể xử lý cả dữ liệu phân loại và liên tục.

#### Hạn chế

* Có thể bị overfitting nếu cây quá phức tạp.
* Nhạy cảm với dữ liệu nhiễu.
* Đôi khi không hiệu quả so với các mô hình khác trong các bài toán phức tạp.

### Random Forest

#### Khái niệm

**Random Forest** là một thuật toán học máy dựa trên **Decision Tree**, được thiết kế để khắc phục một số nhược điểm của Decision Tree, như việc dễ bị overfitting và phụ thuộc quá nhiều vào dữ liệu huấn luyện. Dưới đây là cách **Random Forest** hoạt động dựa trên **Decision Tree**:

#### 2.1.4.2 Decision Tree trong Random Forest

* Một **Decision Tree** là một mô hình học máy chia dữ liệu thành các nhóm nhỏ hơn dựa trên các điều kiện (hoặc thuộc tính) tại mỗi nút trong cây.
* Tuy nhiên, một Decision Tree đơn lẻ có xu hướng dễ bị overfitting, vì nó có thể học quá kỹ vào dữ liệu huấn luyện, đặc biệt khi dữ liệu có nhiễu hoặc kích thước nhỏ.

#### 2.1.4.3 Random Forest: Kết hợp nhiều Decision Tree

Random Forest cải thiện vấn đề của Decision Tree bằng cách kết hợp nhiều cây quyết định trong một hệ thống và đưa ra dự đoán tổng hợp từ tất cả các cây.

#### Nguyên lý chính: "Sức mạnh của tập thể"

* Trong Random Forest, nhiều Decision Tree được tạo ra, và mỗi cây được huấn luyện trên một tập dữ liệu ngẫu nhiên (bagging).
* Các dự đoán từ tất cả các cây được kết hợp:
  + **Phân loại (classification):** Sử dụng **bỏ phiếu đa số** (majority voting) từ tất cả các cây.
  + **Hồi quy (regression):** Sử dụng **trung bình** của các giá trị dự đoán từ các cây.

#### Hai tầng ngẫu nhiên hóa trong Random Forest

Random Forest tăng tính đa dạng giữa các cây bằng cách áp dụng hai loại ngẫu nhiên hóa:

1. **Ngẫu nhiên hóa dữ liệu đầu vào (Bagging):**
   * Mỗi cây được huấn luyện trên một tập con dữ liệu, được lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện ban đầu (có thay thế). Điều này làm cho mỗi cây học từ một phần khác nhau của dữ liệu.
   * Một số mẫu có thể được chọn nhiều lần, trong khi một số khác có thể bị bỏ qua (trung bình, khoảng 2/3 dữ liệu được chọn cho mỗi cây).
2. **Ngẫu nhiên hóa đặc trưng (Random Feature Selection):**
   * Khi tạo một cây, tại mỗi nút phân chia, chỉ một tập hợp con ngẫu nhiên các đặc trưng (features) được xem xét để chọn thuộc tính tốt nhất.
   * Điều này giảm tương quan giữa các cây và giúp mô hình khái quát hóa tốt hơn.

#### Hoạt động của Random Forest dựa trên Decision Tree

1. **Xây dựng các Decision Tree:**
   * Nhiều Decision Tree được xây dựng từ các tập con của dữ liệu và các đặc trưng ngẫu nhiên.
   * Mỗi Decision Tree được huấn luyện độc lập với các cây khác.
2. **Kết hợp dự đoán:**
   * Sau khi xây dựng, Random Forest sử dụng tất cả các cây để đưa ra dự đoán cuối cùng:
     + **Trong phân loại:** Dự đoán cuối cùng là nhãn được đa số các cây bầu chọn.
     + **Trong hồi quy:** Dự đoán cuối cùng là trung bình giá trị dự đoán từ tất cả các cây.

#### Lợi ích của Random Forest dựa trên Decision Tree

1. **Giảm overfitting:**
   * Một Decision Tree đơn lẻ có thể bị overfitting, nhưng Random Forest kết hợp nhiều cây để giảm độ nhạy cảm với nhiễu trong dữ liệu.
2. **Cải thiện khả năng tổng quát hóa:**
   * Việc tổng hợp dự đoán từ nhiều cây giúp Random Forest hoạt động tốt trên dữ liệu mới, không giống như Decision Tree có thể chỉ tốt trên dữ liệu huấn luyện.
3. **Xử lý dữ liệu đa dạng:**
   * Random Forest có thể làm việc tốt với cả dữ liệu phân loại và dữ liệu liên tục (số).

#### Ví dụ minh họa

Giả sử bạn muốn phân loại một loài cây dựa trên chiều cao, đường kính thân, và màu sắc lá.

1. Một Decision Tree sẽ tạo ra các điều kiện cụ thể để phân chia dữ liệu, ví dụ:
   * Nếu chiều cao > 5m và đường kính thân > 10cm thì là "Cây X".
   * Nếu không thì là "Cây Y".
2. **Random Forest** sẽ tạo ra nhiều cây quyết định:
   * Mỗi cây sử dụng một tập dữ liệu ngẫu nhiên và một tập hợp con các đặc trưng.
   * Một cây có thể chỉ xem xét chiều cao và đường kính thân, cây khác có thể chỉ xem xét màu sắc lá.
   * Kết quả cuối cùng là nhãn được đa số các cây đồng ý.

## Cài đặt

### Khai báo các thư viện cần thiết:

A screenshot of a computer program

Description automatically generated

* **numpy**

**NumPy** (Numerical Python) là một thư viện mã nguồn mở trong Python, được sử dụng chủ yếu để thực hiện các phép toán khoa học và số học hiệu quả với các mảng đa chiều và các ma trận. Nó cung cấp các đối tượng mảng n-dimensional, các hàm toán học cho phép thao tác với các mảng này và các công cụ hữu ích khác.

* **pandas**

**Pandas** là một thư viện mã nguồn mở trong Python, chủ yếu được sử dụng để xử lý và phân tích dữ liệu, đặc biệt là với dữ liệu có cấu trúc dạng bảng như dữ liệu trong cơ sở dữ liệu hoặc bảng tính. Pandas cung cấp hai cấu trúc dữ liệu chính là **DataFrame** và **Series**, giúp xử lý và thao tác dữ liệu một cách dễ dàng và hiệu quả.

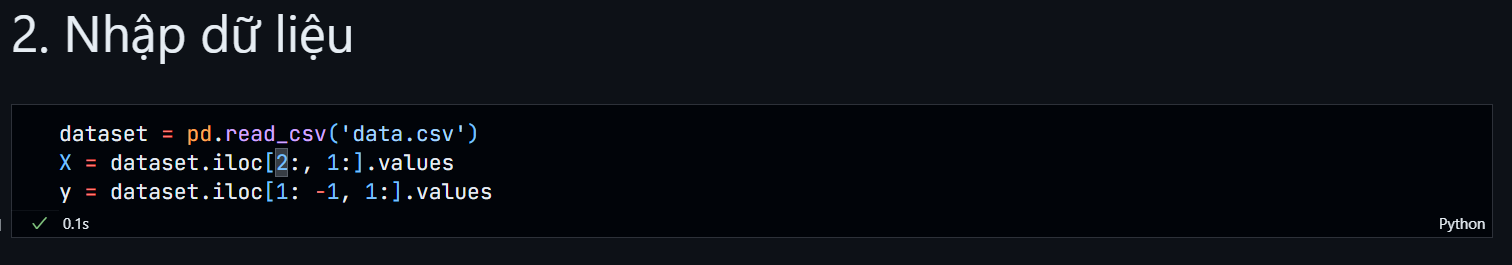
* **sklearn**

**Scikit-learn** (hay thường gọi là **sklearn**) là một thư viện mã nguồn mở trong Python, được sử dụng rộng rãi trong lĩnh vực học máy (Machine Learning). Nó cung cấp các công cụ đơn giản và hiệu quả để phân tích và mô hình hóa dữ liệu, bao gồm các thuật toán học có giám sát và không giám sát, cũng như các công cụ để đánh giá, tối ưu hóa mô hình và tiền xử lý dữ liệu.

* **random**

Thư viện **random** trong Python là một thư viện chuẩn cung cấp các hàm để tạo ra các số ngẫu nhiên, trộn dữ liệu, chọn mẫu ngẫu nhiên từ các tập hợp, và thực hiện nhiều phép toán ngẫu nhiên khác. Đây là một thư viện rất hữu ích trong nhiều ứng dụng, bao gồm mô phỏng, chơi game, thử nghiệm, và sinh dữ liệu ngẫu nhiên.

### Nhập dữ liệu bằng thư viện pandas



Tập X chính là tập dữ liệu thông tin của ngày hôm trước và tập y chính là kết quả của ngày hôm nay, model sẽ dự đoán thông tin xổ số của ngày hôm nay dựa vào dữ liệu của ngày trước.

### Phân chia dữ liệu

A black screen with white text

Description automatically generated

Dữ liệu được phân chia như sau:

**X\_train**: Tập dữ liệu huấn luyện, chứa 80% các đặc trưng từ **X**.

**X\_test**: Tập dữ liệu kiểm tra, chứa 20% các đặc trưng từ **X**.

**y\_train**: Tập nhãn huấn luyện, chứa 80% các nhãn từ **y**.

**y\_test**: Tập nhãn kiểm tra, chứa 20% các nhãn từ **y**.

Chia bộ dữ liệu thành 80% dữ liệu huấn luyện và 20% dữ liệu kiểm tra, và đảm bảo việc chia này được thực hiện một cách nhất quán (dựa vào random\_state). Bộ dữ liệu huấn luyện (**X\_train** và **y\_train**) sẽ được dùng để huấn luyện mô hình, còn bộ dữ liệu kiểm tra (**X\_test** và **y\_test**) sẽ dùng để đánh giá mô hình sau khi huấn luyện.

### Huấn luyện các mô hình

#### Mô hình Linear Regression

A black and white text

Description automatically generated

* Khởi tạo mô hình hồi quy tuyến tính với LinearRegression().
* Huấn luyện mô hình với dữ liệu huấn luyện X\_train và y\_train bằng phương thức fit().

#### Mô hình Polynominal Regression

A screenshot of a computer

Description automatically generated

* Sử dụng PolynomialFeatures để chuyển đổi các đặc trưng trong X\_train thành các đặc trưng đa thức với bậc 2 (bao gồm x, x2).
* Sau khi biến đổi dữ liệu, sử dụng mô hình hồi quy tuyến tính (LinearRegression) để huấn luyện mô hình với các đặc trưng đa thức (X\_poly) và nhãn (y\_train).

#### Mô hình Decision Tree

A screenshot of a computer

Description automatically generated

* Khởi tạo mô hình cây quyết định hồi quy (DecisionTreeRegressor) với tham số random\_state=0 để đảm bảo tính tái tạo của mô hình.
* Huấn luyện mô hình với dữ liệu huấn luyện: Dữ liệu đầu vào X\_train và nhãn y\_train được sử dụng để xây dựng cây quyết định.

#### Mô hình Random Forest

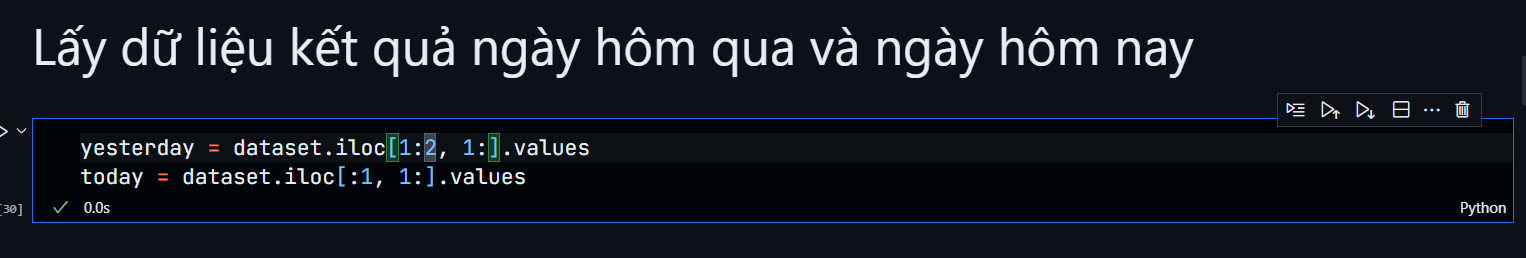
A screenshot of a computer

Description automatically generated

* Khởi tạo mô hình rừng quyết định hồi quy với 1000 cây quyết định (n\_estimators=1000), và đảm bảo kết quả tái tạo được với random\_state=0.
* Huấn luyện mô hình với dữ liệu huấn luyện X\_train (đặc trưng) và y\_train (nhãn).

### Sử dụng các mô hình để dự đoán kết quả xổ số hôm nay

#### Lấy dữ liệu kết quả của ngày hôm qua và ngày hôm này



* yesterday sẽ lưu kết quả xổ số của ngày hôm qua
* today sẽ lưu kết quả xố số của ngày hôm nay

#### Cho các mô hình dự đoán kết quả xổ số hôm này

A screen shot of a computer code

Description automatically generated

* prediction1 là kết quả dự đoán của mô hình Linear Regression
* prediction2 là kết quả dự đoán của mô hình Polynominal Regression
* prediction3 là kết quả dự đoán của mô hình Decision Tree Regression
* prediction4 là kết quả dự đoán của mô hình Random Forest Regression

#### Làm tròn lại các số

A black rectangular object with a black stripe

Description automatically generated

#### In ra kết quả dự đoán

A screenshot of a computer

Description automatically generated

* So sánh với kết quả xổ số hôm nay

Kết quả xổ số hôm nay

A black background with white text

Description automatically generated

So sánh:

A screenshot of a computer program

Description automatically generated

### Đánh giá chất lượng các mô hình

A screen shot of a computer

Description automatically generated

* y\_pred1 là dự đoán từ mô hình Linear Regression.
* y\_pred2 là dự đoán từ mô hình Polynominal Regression.
* y\_pred3 là dự đoán từ mô hình Decision Tree Regression.
* y\_pred4 là dự đoán từ mô hình Random Forest Regression.
* So sánh kết quả dự đoán với kết quả thực tế (y\_test)

A black rectangular object with a black stripe

Description automatically generated

R2 score là một chỉ số dùng để đánh giá mức độ phù hợp của mô hình với dữ liệu. R² cho biết phần trăm biến thiên trong biến mục tiêu (y) có thể được giải thích bởi các biến đầu vào (X) trong mô hình. Ở đây chúng ta thấy R2 score ở cả 4 mô hình đều bé hơn 0. Điều này có nghĩa là các mô hình đang hoạt động không tốt và khó có thể dự đoán được kết quả xổ số dựa trên thông tin của những ngày trước.

# Kết Luận

Chương trình dự đoán kết quả xổ số là một ứng dụng sử dụng các phương pháp thống kê, thuật toán học máy, hoặc phân tích dữ liệu để phân tích và đưa ra các con số có khả năng xuất hiện cao dựa trên dữ liệu quá khứ. Tuy nhiên, do tính chất ngẫu nhiên và không thể đoán trước của xổ số, mọi dự đoán chỉ mang tính tham khảo, không đảm bảo độ chính xác tuyệt đối.

Mặc dù các thuật toán hiện đại có thể cải thiện khả năng phân tích dữ liệu, người chơi cần nhận thức rằng xổ số là một trò chơi may rủi, và việc dựa vào các dự đoán không nên dẫn đến các hành động đầu tư quá mức hoặc kỳ vọng không thực tế.

Nhìn chung, chương trình dự đoán kết quả xổ số có thể là một công cụ thú vị, hỗ trợ người chơi trong việc chọn số, nhưng quan trọng hơn hết là chơi xổ số một cách có trách nhiệm và coi đó như một hình thức giải trí.