## BỘ GIÁO DỤC VÀ ĐÀO TẠO



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

BÀI TẬP LỚN

**HỌC MÁY CƠ BẢN**

**APPLYING MACHINE LEARING TO PREDICT STROKE**

NGÀNH: CÔNG NGHỆ PHẦN MỀM

SINH VIÊN: **1. NGUYỄN VĂN LINH**

**2. ĐÀO ANH QUÂN**

LỚP: **12522W.2**

NGƯỜI HƯỚNG DẪN: **TS. HOÀNG QUỐC VIỆT**

**HƯNG YÊN – 2025**

# NHẬN XÉT

## Nhận xét của giáo viên hướng dẫn

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

...........................................................................................................................................

## GIÁO VIÊN HƯỚNG DẪN

**Hoàng Quốc Việt**

**MỤC LỤC**

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 6](#_bookmark0)

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 9](#_bookmark1)

* 1. [Bài toán 9](#_bookmark2)
  2. [Trình bày dữ liệu bài toán 9](#_bookmark3)
  3. [Tiền xử lý dữ liệu 11](#_bookmark7)
  4. [Làm sạch dữ liệu 12](#_bookmark8)
  5. [Trực quan hoá dữ liệu 12](#_bookmark9)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 13](#_bookmark10)

* 1. [RandomForestClassifier 13](#_bookmark11)
     1. [Giới thiệu thuật toán RandomForestClassifier 13](#_bookmark12)
     2. [Cấu trúc của mô hình RandomForest 13](#_bookmark13)
     3. [Quá trình xây dựng mô hình RandomForest 13](#_bookmark14)
     4. [Ưu điểm của RandomForest 13](#_bookmark15)
     5. [Nhược điểm của RandomForest 14](#_bookmark16)
     6. [Đánh giá hiệu suất 14](#_bookmark17)
  2. [KNeighborsClassifier 14](#_bookmark18)
     1. [Giới thiệu thuật toán KNeighborsClassifier 14](#_bookmark19)
     2. [Cấu trúc của mô hình KNN 14](#_bookmark20)
     3. [Quá trình xây dựng mô hình KNN 15](#_bookmark21)
     4. [Ưu điểm của KNN 15](#_bookmark22)
     5. [Nhược điểm của KNN 15](#_bookmark23)
     6. [Đánh giá hiệu suất 15](#_bookmark24)
  3. [Support Vector Classification (SVC) 15](#_bookmark25)
     1. [Giới thiệu thuật toán SVC 15](#_bookmark26)
     2. [Cấu trúc của mô hình SVC 16](#_bookmark27)
     3. [Quá trình xây dựng mô hình SVC 16](#_bookmark28)
     4. [Ưu điểm của SVC 16](#_bookmark29)
     5. [Nhược điểm của SVC 16](#_bookmark30)
     6. [Đánh giá hiệu suất 16](#_bookmark31)

[CHƯƠNG 3: GIẢI PHÁP, CÀI ĐẶT THỰC HIỆN 17](#_bookmark32)

* 1. [Mã nguồn tiền xử lý dữ liệu 17](#_bookmark33)
  2. [Mã nguồn chức năng làm sạch dữ liệu 18](#_bookmark38)
     1. [Xử lý giá trị thiếu 18](#_bookmark39)
     2. [Xoá outlier 20](#_bookmark46)
     3. [Ma trận tương quan 21](#_bookmark49)
  3. [Mã nguồn chức năng Trực quan hoá dữ liệu 23](#_bookmark52)
     1. [Biểu đồ thể hiện phân phối đặc trưng các phân loại 23](#_bookmark53)
     2. [Biểu đồ thể hiện đếm số lần đột quỵ 24](#_bookmark57)
     3. [Biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark60)
     4. [Biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại 26](#_bookmark63)
     5. [Biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age,](#_bookmark66) [avg\_glucose\_level) 27](#_bookmark66)
     6. [Biểu đồ kiểm tra giá trị ngoại lai 27](#_bookmark69)
     7. [Biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark72)
  4. [Chuyển đổi dữ liệu 29](#_bookmark75)
     1. [Xử lý dữ liệu chuỗi 29](#_bookmark77)
     2. [Xử lý dữ liệu số 30](#_bookmark81)
     3. [Xử lý dữ liệu mất cân bằng bằng SMOTE 31](#_bookmark85)
  5. [Huấn luyện mô hình SKLEARN 31](#_bookmark87)
     1. [Tìm kiếm mô hình 31](#_bookmark88)
     2. [Điều chỉnh tham số 32](#_bookmark91)
     3. [Mô hình RandomForestClassifier 35](#_bookmark95)
     4. [Mô hình KNeighborsClassifier 36](#_bookmark98)
     5. [Mô hình Support Vector Classification 37](#_bookmark101)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG 38](#_bookmark104)

[KẾT LUẬN 39](#_bookmark106)

[TÀI LIỆU THAM KHẢO 41](#_bookmark107)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1. 1 Hiển thị 5 hàng đầu và 5 hàng cuối của dữ liệu 10](#_bookmark4)

[Hình 1. 2 Thông tin tổng quan về dữ liệu 10](#_bookmark5)

[Hình 1. 3 Thông tin mô tả về DataFrame 11](#_bookmark6)

[Hình 3. 1 Import thư viện cần thiết, load, hiển thị 5 hàng đầu của DataFrame 17](#_bookmark34)

[Hình 3. 2 Kết quả hiển thị 5 hàng đầu của DataFrame 17](#_bookmark35)

[Hình 3. 3 Mã nguồn xoá cột “id” 17](#_bookmark36)

[Hình 3. 4 Thông tin về DataFrame 18](#_bookmark37)

[Hình 3. 5 Mã nguồn xử lý giá trị thiếu 18](#_bookmark40)

[Hình 3. 6 Hiển thị phần trăm giá trị thiếu của từng cột 19](#_bookmark41)

[Hình 3. 7 Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu](#_bookmark42) [(NaN) và hiển thị 5 hàng cuối của DataFrame 19](#_bookmark42)

[Hình 3. 8 Kết quả hiển thị 5 hàng cuối của DataFrame 19](#_bookmark43)

[Hình 3. 9 Mã nguồn đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame](#_bookmark44) [data. 20](#_bookmark44)

[Hình 3. 10 Kết quả liệt kê số giá trị thiếu theo cột. 20](#_bookmark45)

[Hình 3. 11 Mã nguồn xử lý dữ liệu ngoại lai 20](#_bookmark47)

[Hình 3. 12 Boxplot minh họa dữ liệu sạch. 21](#_bookmark48)

[Hình 3. 13 Mã nguồn ma trận tương quan 21](#_bookmark50)

[Hình 3. 14 Kết quả Heatmap minh họa mức độ tương quan giữa các yếu tố 22](#_bookmark51)

[Hình 3. 15 Mã nguồn biểu đồ thể hiện phân phối đặc trưng các phân loại 23](#_bookmark54)

[Hình 3. 16 Kết quả biểu đồ bar thể hiện phân phối đặc trưng các phân loại 23](#_bookmark55)

[Hình 3. 17 Kết quả biểu đồ pie thể hiện phân phối đặc trưng các phân loại 24](#_bookmark56)

[Hình 3. 18 Mã nguồn biểu đồ thể hiện đếm số lần đột quỵ 24](#_bookmark58)

[Hình 3. 19 Kết quả biểu đồ thể hiện đếm số lần đột quỵ 25](#_bookmark59)

[Hình 3. 20 Mã nguồn biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark61)

[Hình 3. 21 Kết quả biểu đồ thể hiện các đặc trưng liên tục 25](#_bookmark62)

[Hình 3. 22 Mã nguồn biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn](#_bookmark64) [lại 26](#_bookmark64)

[Hình 3. 23 Kết quả biểu đồ thể hiện mối tương quan giữa đột quỵ và các phần tử còn lại](#_bookmark65)

[. 26](#_bookmark65)

[Hình 3. 24 Mã nguồn biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho](#_bookmark67) [(age, avg\_glucose\_level) 27](#_bookmark67)

[Hình 3. 25 Kết quả biểu đồ thể hiện sự phân bố đột quỵ và không đột quỵ dành cho (age,](#_bookmark68) [avg\_glucose\_level) 27](#_bookmark68)

[Hình 3. 26 Mã nguồn biểu đồ kiểm tra giá trị ngoại lai 27](#_bookmark70)

[Hình 3. 27 Kết quả biểu đồ kiểm tra giá trị ngoại lai 28](#_bookmark71)

[Hình 3. 28 Mã nguồn biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark73)

[Hình 3. 29 Kết quả biểu đồ thể hiện sự trực quan hoá giá trị thiếu 28](#_bookmark74)

[Hình 3. 30 Import 1 số thư viện cần thiết 29](#_bookmark76)

[Hình 3. 31 Mã nguồn xử lý dữ liệu chuỗi 29](#_bookmark78)

[Hình 3. 32 Kết quả xử lý dữ liệu chuỗi 29](#_bookmark79)

[Hình 3. 33 Mã nguồn xoá cột phân loại gốc trong X\_train và X\_test 30](#_bookmark80)

[Hình 3. 34 Mã nguồn xử lý dữ liệu số 30](#_bookmark82)

[Hình 3. 35 Kết quả xử lý dữ liệu số 30](#_bookmark83)

[Hình 3. 36 Hiển thị số hàng và số cột có X\_train và y\_train 30](#_bookmark84)

[Hình 3. 37 Xử lý dữ liệu mất cân bằng bằng SMOTE 31](#_bookmark86)

[Hình 3. 38 Mã nguồn tìm kiếm mô hình dùng Lazypredict 31](#_bookmark89)

[Hình 3. 39 Kết quả tìm kiếm mô hình dùng Lazypredict 32](#_bookmark90)

[Hình 3. 40 Import các thư viện cần thiết 32](#_bookmark92)

[Hình 3. 41 Mã nguồn hàm tìm kiếm, điều chỉnh tham số tốt nhất 33](#_bookmark93)

[Hình 3. 42 Mã nguồn trực quan hoá, hiển thị kết quả sau khi chạy mô hình 34](#_bookmark94)

[Hình 3. 43 Mã nguồn mô hình RandomForestClassifier 35](#_bookmark96)

[Hình 3. 44 Kết quả mô hình RandomForestClassifier 35](#_bookmark97)

[Hình 3. 45 Mã nguồn mô hình KneighborsClassifier 36](#_bookmark99)

[Hình 3. 46 Kết quả mô hình KNeighborsClassifier 36](#_bookmark100)

[Hình 3. 47 Mã nguồn mô hình Support Vector Classification 37](#_bookmark102)

[Hình 3. 48 Kết quả mô hình Support Vector Classification 37](#_bookmark103)

[HÌnh 4. 1 Sử dụng Streamlit để đưa bài toán lên giao diện web 38](#_bookmark105)

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## Bài toán

Tổng sản phẩm quốc nội(GDP) là một trong những chỉ số quan trọng phản ánh mức độ phát triển của một quốc gia. Dự đoán chính xác GDP thông qua việc phân tích dữ liệu và áp dụng các thuật toán học máy có thể hỗ trợ các chính phủ, tổ chức và doanh nghiệp xây dựng chính sách kinh tế, kế hoạch đầu tư và các chiến lược phát triển bển vững.

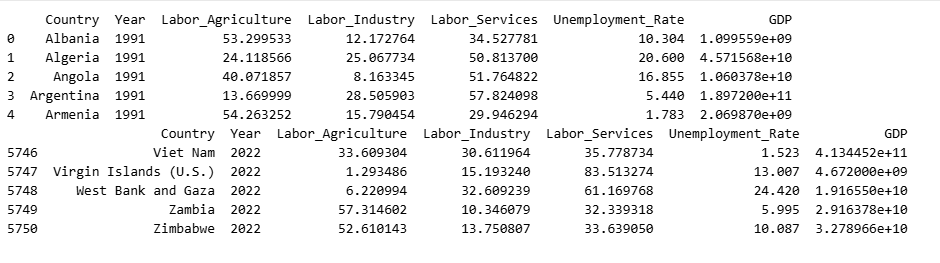
Bài toán "Áp dụng các thuật toán học máy vào dự đoán GDP dựa trên dữ liệu cơ cấu lao động và tỷ lệ thất nghiệp" tập trung vào việc khai thác dữ liệu kinh tế - xã hội và áp dụng các thuật toán học máy để xây dựng mô hình dự báo GDP. Từ đó, chúng ta sẽ xây dựng các mô hình dự báo sử dụng các kỹ thuật học máy như hồi quy logistic, rừng ngẫu nhiên, và máy vector hỗ trợ (SVM) và các bộ dữ liệu được sử dụng bao gồm các thông tin cơ bản như: tên quốc gia, năm, tỷ lệ lao động trong nông nghiệp, tỷ lệ lao động trong công nghiệp, tỷ lệ lao động trong dịch vụ, tỷ lệ thất nghiệp .

Dựa vào tích dữ liệu GDP từng năm của các quốc gia, các mô hình học máy có thể phát hiện các mối quan hệ tiềm ẩn giữa cơ cấu lao động và tốc độ tang trưởng GDP. Từ đó, nghiên cứu hướng tới việc:

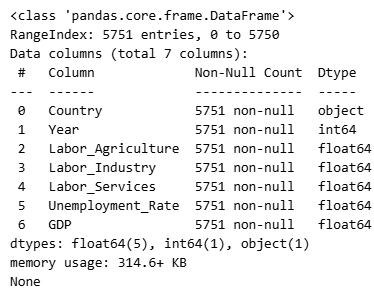
* + - Xác định mức độ ảnh hưởng của từng khu vực lao động đến tăng trưởng GDP.
    - Đánh giá vai trò của tỷ lệ thất nghiệp trong việc phản ánh tiềm năng phát triển của quốc gia.
    - Xây dựng so sách các mô hình học máy có độ chính xác cao trong dự đoán GDP hỗ trợ các chính sách và chiến lược phát triển.

Đề tài không chỉ thể hiện giá trị thực tiễn của học máy trong lĩnh vực kinh tế mà còn mang ý nghĩa khoa học sâu sắc trong việc sử dụng trí tuệ nhân tạo vào phân tích, dự báo và hỗ trợ ra quyết định. Kết quả nghiên cứu có thể góp phần nâng cao hiệu quả điều hành kinh tế, giúp các quốc gia ra chính sách, giải pháp hợp lý nhằm ổn định và phát triển kinh tế một cách bền vững.

* 1. **Trình bày dữ liệu bài toán** Dữ liệu được lấy từ trang web này: [Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)



*Hình 1. 1 Hiển thị 5 hàng đầu và 5 hàng cuối của dữ liệu*

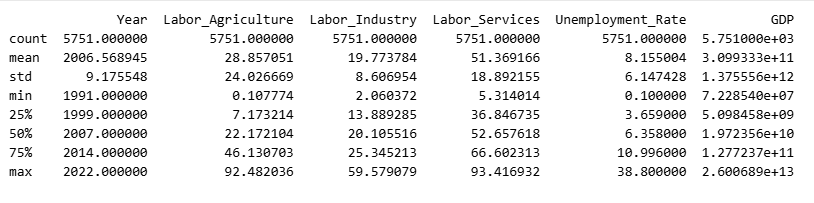
**

*Hình 1. 2 Thông tin tổng quan về cấu trúc Dataframe*

* Bộ dữ liệu được sử dụng trong bài toán này bao gồm các đặc trưng liên quan đến sức khỏe và nhân khẩu học của bệnh nhân, cụ thể như sau:
  1. Country Name: tên quốc gia .
  2. Year: năm quan sát.
  3. Employment Sector: Agriculture: Tỷ lệ lao động ngành Nông nghiệp.
  4. Employment Sector: Industry: Tỷ lệ lao động ngành Công nghiêp.
  5. Employment Sector: Services: Tỷ lệ lao động ngành dịch vụ
  6. Unemployment Rate: Tỷ lệ thất nghiệp
  7. GDP (in USD): Tổng sản phẩm quốc nội tính theo USD
* Dữ liệu bài toán là 1 file csv gồm 5751 rows × 7 columns

+ Có 7 feature và mỗi feature có 5751 dữ liệu đầu vào

-Sau khi mô tả dữ liệu ta có:



*Hình 1. 3 Thông tin mô tả về dữ liệu số*

+ Năm thống kê lớn nhất là **2022** và nhỏ nhất là **1991**.

+ Tỷ lệ lao động trong lĩnh vực **nông nghiệp (Agriculture)** lớn nhất là **92.48%** và nhỏ nhất là **0.11%**.

+ Tỷ lệ lao động trong lĩnh vực **công nghiệp (Industry)** lớn nhất là **59.58%** và nhỏ nhất là **2.06%**.

+ Tỷ lệ lao động trong lĩnh vực **dịch vụ (Services)** lớn nhất là **93.42%** và nhỏ nhất là **5.31%**.

+ **Tỷ lệ thất nghiệp (Unemployment Rate)** lớn nhất là **38.80%** và nhỏ nhất là **0.10%.**

+ **GDP (tính theo USD)** lớn nhất là **2.60×10¹³ USD** và nhỏ nhất là **7.23×10⁷ USD.**

## Tiền xử lý dữ liệu

1. Hiển thị 5 hàng đầu của dữ liệu
2. Hiển thị số hàng, cột của dữ liệu
3. Hiển thị thông tin của dữ liệu
4. Mô tả dữ liệu
5. Hiển thị giá trị duy nhất trong mỗi cột

## Làm sạch dữ liệu

1. Xử lý giá trị thiếu
2. Xoá outliner
3. Ma trận tương quan

## Trực quan hoá dữ liệu

1. Biểu đồ thể hiện phân phối đặc trưng các phân loại
2. Biểu đồ phân bố GDP
3. đồ ma trận tương quan
4. Biểu đồ phân tán GDP theo tỷ lệ lao động nông nghiệp
5. Biểu đồ phân tán GDP theo tỷ lệ lao động công nghiệp
6. **Biểu đồ phân tán GDP theo tỷ lệ lao động dịch vụ**
7. Biểu đồ hộp về tỷ lệ thất nghiệp

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## RandomForestClassifier

## Giới thiệu thuật toán RandomForestClassifier

Thuật toán RandomForestClassifier là một phương pháp học máy phổ biến được sử dụng trong cả bài toán hồi quy. Random Forest là một tập hợp (Ensemble model) các cây quyết định (Decision Tree), trong đó mỗi cây được xây dựng dựa trên một tập dữ liệu mẫu ngẫu nhiên (Bootstrapping) và một tập con các đặc trưng (Feature subset) được chọn ngẫu nhiên tại mỗi nút. Kết quả dự đoán cuối cùng được lấy bằng **trung bình giá trị dự đoán của các cây thành phần**, giúp tăng độ chính xác và giảm hiện tượng overfitting.

## Cấu trúc của mô hình RandomForest

Mô hình Random Forest bao gồm:

* + - * **Cây quyết định (Decision Trees):** Các mô hình con học các mối quan hệ phi tuyến giữa đặc trưng và biến mục tiêu.
      * **Tập dữ liệu mẫu (Bootstrap Samples):** Mỗi cây được huấn luyện trên một mẫu ngẫu nhiên có thay thế từ tập dữ liệu gốc.
      * **Tập con đặc trưng (Feature Subset):** Ở mỗi nút, chỉ một số đặc trưng được chọn ngẫu nhiên để tìm điểm chia tối ưu.

## Quá trình xây dựng mô hình RandomForest

* + - * **Khởi tạo các cây :** Xác định số lượng cây (tham số n\_estimators)
      * **Lấy mẫu dữ liệu:** Mỗi cây được huấn luyện trên tập con dữ liệu riêng.
      * **Chọn tập con đặc trưng:** Mỗi nút chia dữ liệu dựa trên tập con đặc trưng được chọn ngẫu nhiên.
      * **Xây dựng cây:** Từng cây được huấn luyện độc lập, sử dụng chỉ tiêu như “Mean Squared Error” để tối ưu việc chia.
      * **Dự đoán:** Kết quả cuối cùng là **trung bình dự đoán của toàn bộ cây** trong rừng.

## Ưu điểm của RandomForest

* + - * **Khả năng tổng quát hóa tốt:** Kết hợp nhiều cây giúp mô hình ổn định hơn và tổng quát hóa tốt.
      * **Hiệu suất cao:** Làm việc hiệu quả với dữ liệu lớn, phi tuyến.
      * **Đánh giá quan trọng đặc trưng:** Cho biết yếu tố nào (ví dụ: tỷ lệ lao động nông nghiệp, công nghiệp, dịch vụ, thất nghiệp) ảnh hưởng mạnh nhất đến GDP.
      * **Độ chính xác cao:** Thường cho kết quả chính xác hơn mô hình hồi quy tuyến tính đơn giản.
      * **Xử lý dữ liệu mất mát:** Có khả năng ước lượng giá trị bị thiếu trong dữ liệu.

## Nhược điểm của RandomForest

* + - * **Khó giải thích:** Mặc dù mỗi cây dễ hiểu, nhưng tổng thể rừng lại phức tạp.
      * **Chi phí tính toán cao:** Huấn luyện nhiều cây đòi hỏi tài nguyên lớn.
      * **Dễ bị bias:** Nếu dữ liệu không được chuẩn hóa hoặc phân bố không đồng đều.

## Đánh giá hiệu suất

Hiệu suất của RandomForestClassifier thường được đánh giá thông qua các chỉ số như Accuracy, MAE (Mean Absolute Error, RMSE (Root Mean Squared Error), và **R² (Coefficient of Determination)**. Đối với bài toán dự đoán GDP , mô hình được kiểm tra dựa trên khả năng dự đoán chính xác GDP thực tế của các quốc gia trong giai đoạn quan sát.

## KNeighborsClassifier

## Giới thiệu thuật toán KNeighborsClassifier

Thuật toán **K-Nearest Neighbors (KNN)** là một phương pháp học máy **phi tham số (non-parametric)**, có thể được sử dụng cho cả **phân loại (classification)** và **hồi quy (regression)**. Trong bài toán hồi quy, mô hình **KNeighborsRegressor** dự đoán giá trị đầu ra của một điểm dữ liệu mới dựa trên **trung bình giá trị mục tiêu (GDP)** của **k điểm lân cận gần nhất** trong tập huấn luyện. KNN dựa trên nguyên tắc: “Các quốc gia có đặc trưng kinh tế giống nhau (cấu trúc lao động, tỷ lệ thất nghiệp tương đồng) sẽ có giá trị GDP gần nhau.”

## Cấu trúc của mô hình KNN

* + - * **Tập huấn luyện:** Gồm các điểm dữ liệu có giá trị GDP đã biết.
      * **Khoảng cách:** Các thước đo như **Euclidean**, **Manhattan**, hoặc **Minkowski** được dùng để tính độ tương đồng giữa các quốc gia.
      * **Số lượng lân cận (k):** là siêu tham số chính, xác định bao nhiêu điểm lân cận sẽ được sử dụng trong quá trình dự đoán.

## Quá trình xây dựng mô hình KNN

* + - * **Xác định giá trị k:** Thường chọn k là số lẻ hoặc dựa trên cross-validation.
      * **Tính khoảng cách:** Tính khoảng cách giữa quốc gia cần dự đoán và toàn bộ các quốc gia trong tập huấn luyện.
      * **Chọn k lân cận gần nhất:** Lấy k điểm gần nhất theo khoảng cách đã chọn.
      * **Bỏ phiếu:** Giá trị GDP dự đoán là **trung bình GDP của k quốc gia gần nhất**.

## Ưu điểm của KNN

* + - * **Đơn giản:** Không cần huấn luyện phức tạp.
      * **Linh hoạt:** Có thể xử lý dữ liệu tuyến tính hoặc phi tuyến.
      * **Không cần huấn luyện:** Khi số lượng quốc gia trong tập huấn luyện không quá lớn.
      * **Hiệu quả với dữ liệu nhỏ:** Khi các biến đầu vào đa dạng như lao động, thất nghiệp, năm, v.v.

## Nhược điểm của KNN

* + - * **Chi phí tính toán cao:** Khi tập dữ liệu lớn, cần tính khoảng cách với tất cả các điểm.
      * **Nhạy cảm với dữ liệu nhiễu:** Các giá trị bất thường (outlier) có thể làm sai lệch dự đoán GDP.
      * **Phụ thuộc vào k:** Việc chọn giá trị k không phù hợp có thể dẫn đến sai số cao.
      * **Cần chuẩn hóa dữ liệu:** Vì các đặc trưng có thể có thang đo khác nhau.

## Đánh giá hiệu suất

Hiệu suất của mô hình KNeighborsRegressor được đánh giá qua các chỉ số: **MAE**, **RMSE**, và **R²**. Trong bài toán GDP, mô hình KNN giúp so sánh một quốc gia mới với các quốc gia có đặc trưng kinh tế tương đồng, tuy nhiên độ chính xác phụ thuộc nhiều vào việc lựa chọn k và chuẩn hóa dữ liệu.

## Support Vector Regression (SVR)

## Giới thiệu thuật toán SVR

**Support Vector Regression (SVR)** là biến thể hồi quy của **Support Vector Machine (SVM),** được thiết kế để giải quyết bài toán **dự đoán biến liên tục.**  
SVR tìm một **hàm hồi quy tối ưu (optimal hyperplane)** sao cho sai số giữa giá trị dự đoán và giá trị thực **nằm trong một giới hạn cho phép (epsilon)**, đồng thời **biên (margin)** giữa các điểm dữ liệu được tối đa hóa.

## Cấu trúc của mô hình SVR

* + - * **Siêu phẳng (Hyperplane):** Là đường hoặc mặt phẳng biểu diễn mối quan hệ giữa đặc trưng và GDP.
      * **Margin:** Khoảng cách cho phép sai số không bị phạt.
      * **Hạt nhân (Kernel):** Ánh xạ dữ liệu sang không gian cao hơn để mô hình hóa mối quan hệ phi tuyến (ví dụ: RBF, Polynomial, Sigmoid).

## Quá trình xây dựng mô hình SVR

* + - * **Lựa Chọn kernel:** Thường sử dụng **RBF kernel** vì phù hợp với dữ liệu kinh tế phi tuyến.
      * **Tìm siêu phẳng tối ưu:** Sử dụng tham số C để điều chỉnh mức phạt và ε để xác định ngưỡng sai số cho phép.
      * **Đánh giá:** Mô hình được huấn luyện để giảm sai số và dự đoán GDP của quốc gia mới.

## Ưu điểm của SVR

* + - * **Hiệu quả cao:** Là một trong những thuật toán tốt nhất cho các bài toán phân loại nhị phân.
      * **Giảm overfitting:** Nhờ cơ chế margin tối ưu..
      * **Khả năng tổng quát tốt:** Làm việc tốt khi dữ liệu có phân bố phức tạp.
      * **Hiệu quả với dữ liệu nhỏ:** Hoạt động tốt ngay cả khi kích thước dữ liệu nhỏ.

## Nhược điểm của SVR

* + - * **Thời gian huấn luyện lâu:** Khi dữ liệu lớn hoặc nhiều đặc trưng.
      * **Cần tinh chỉnh tham số:** Việc chọn C, ε, và kernel ảnh hưởng lớn đến hiệu suất.
      * **Khó diễn giải:** Mô hình hoạt động như “hộp đen”, khó hiểu rõ quan hệ bên trong.

## Đánh giá hiệu suất

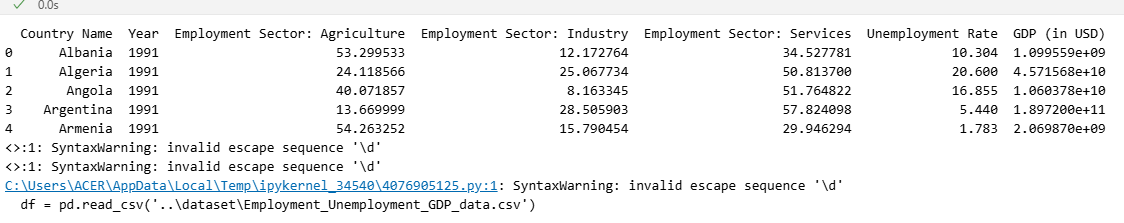
Hiệu suất của mô hình SVR được đánh giá bằng các chỉ số **MAE**, **RMSE**, và **R²**.  
Trong bài toán dự đoán GDP, SVR có khả năng **mô hình hóa mối quan hệ phi tuyến** giữa GDP và các yếu tố như tỷ lệ lao động, thất nghiệp, và năm, cho kết quả ổn định ngay cả khi dữ liệu huấn luyện không lớn.

# CHƯƠNG 3: GIẢI PHÁP, CÀI ĐẶT THỰC HIỆN

## Mã nguồn tiền xử lý dữ liệu

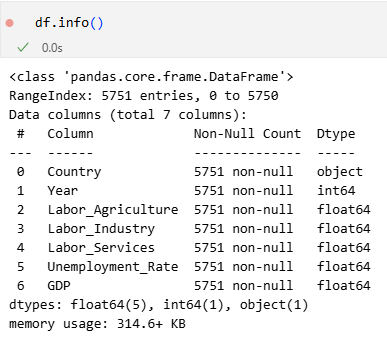
****

*Hình 3. 1 Import thư viện cần thiết, load, hiển thị 5 hàng đầu của DataFrame*

**

*Hình 3. 2 Kết quả hiển thị 5 hàng đầu của DataFrame*

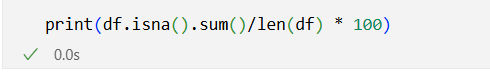
Đầu tiên em sử dụng hàm df.info() để kiểm tra các feature nào bị thiếu và có kiểu dữ liệu thế nào.Em hiển thị lại hàm df và dùng lại hàm info để kiểm tra xem dữ liệu. Và hình ảnh dưới là kết quả sau khi đã điền các dữ liệu:



*Hình 3. 4 Thông tin về DataFrame*

## Mã nguồn chức năng làm sạch dữ liệu

## Xử lý giá trị thiếu

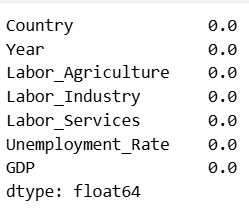
****

*Hình 3. 5 Mã nguồn xử lý giá trị thiếu*

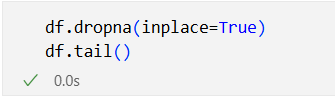
Câu lệnh này tính toán tỷ lệ phần trăm giá trị thiếu trong từng cột của DataFrame data.

* + - * **df.isna()**: Xác định các giá trị thiếu (NaN) trong data và trả về DataFrame với giá trị True/False.
      * **.sum()**: Tính tổng số True (giá trị thiếu) cho mỗi cột.
      * **len(data)**: Số lượng hàng trong data.
      * **\*100**: Chuyển đổi tỷ lệ sang phần trăm.

Kết quả: Hiển thị phần trăm giá trị thiếu của từng cột, hỗ trợ xác định cần xử lý những cột nào.



*Hình 3. 6 Hiển thị phần trăm giá trị thiếu của từng cột*

**

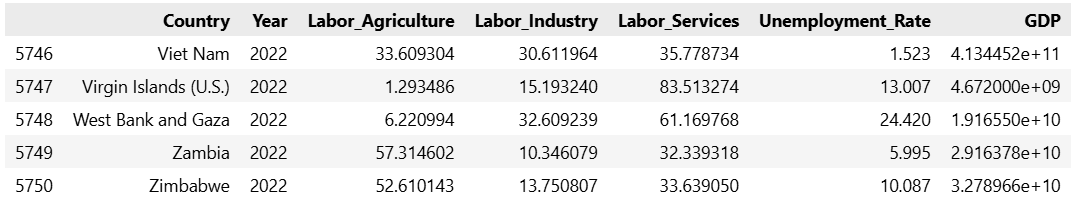
*Hình 3. 7 Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu (NaN) và hiển thị 5 hàng cuối của DataFrame*

Câu lệnh này được sử dụng để xử lý giá trị thiếu và kiểm tra dữ liệu sau khi loại bỏ các hàng chứa giá trị thiếu:

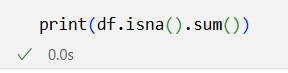
**df.dropna(inplace=True)**: Loại bỏ tất cả các hàng trong DataFrame data có ít nhất một giá trị bị thiếu (NaN). Tham số inplace=True đảm bảo thay đổi được thực hiện trực tiếp trên DataFrame data mà không tạo một bản sao.

**df.tail()**: Hiển thị 5 hàng cuối cùng của DataFrame data sau khi loại bỏ các giá trị thiếu.

Mục đích: Loại bỏ dữ liệu không đầy đủ và xem nhanh các dòng cuối để đảm bảo dữ liệu đã được xử lý đúng cách.



*Hình 3. 8 Kết quả hiển thị 5 hàng cuối của DataFrame*



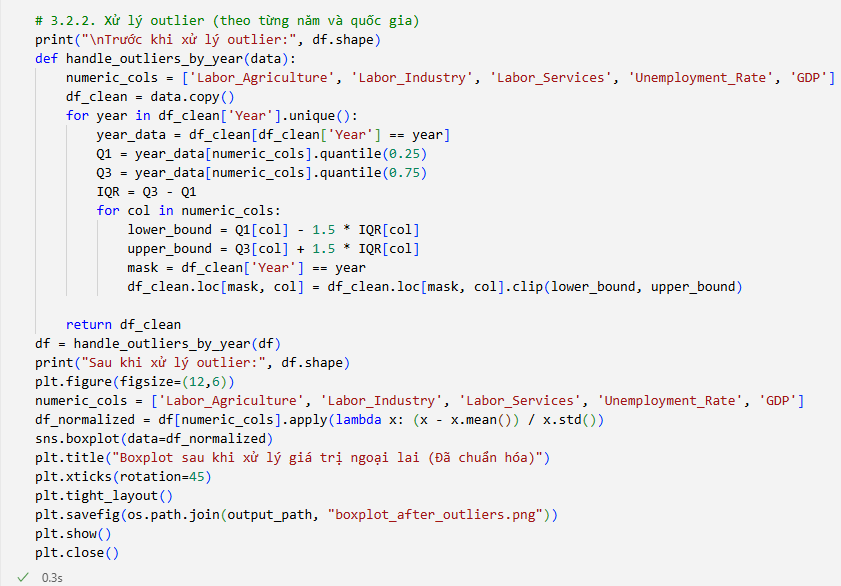
*Hình 3. 9 Mã nguồn đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame data.*

* + - * Đếm số lượng giá trị thiếu (NaN) trong từng cột của DataFrame data.
      * Kết quả: Một bảng liệt kê số giá trị thiếu theo cột.



*Hình 3. 10 Kết quả liệt kê số giá trị thiếu theo cột.*

## Xoá outlier

****

*Hình 3. 11 Mã nguồn xử lý dữ liệu ngoại lai*

Hàm **handle\_outliers** loại bỏ giá trị ngoại lai trong các cột bằng cách sử dụng **IQR (Interquartile Range)**:

## Tính IQR:

* + Q1 (25%) và Q3 (75%) là các phân vị của dữ liệu.
  + IQR = Q3 - Q1, thể hiện khoảng giữa 50% dữ liệu trung tâm.

## Giới hạn:

* + **Lower bound** = Q1 - 1.5 \* IQR.
  + **Upper bound** = Q3 + 1.5 \* IQR.

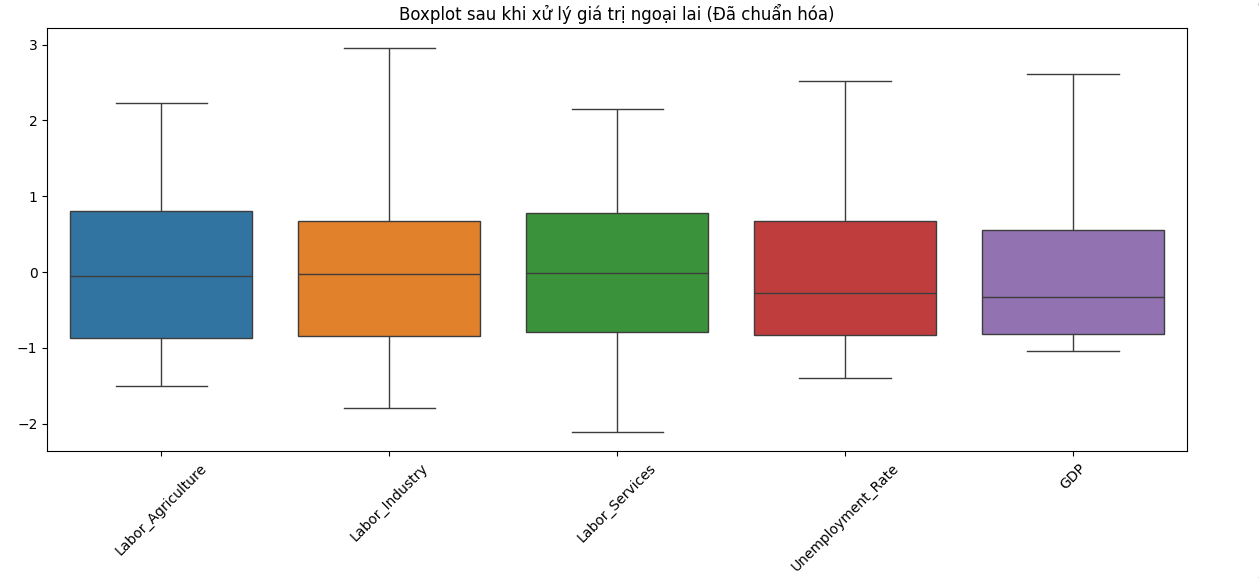
## Lọc dữ liệu:

* + Loại bỏ các dòng có giá trị nằm ngoài khoảng [Lower bound, Upper bound] ở bất kỳ cột số nào
  + Sau khi lọc, dữ liệu chỉ còn lại các giá trị nằm trong phạm vi hợp lý, đảm bảo tính ổn định cho quá trình huấn luyện.

## boxplots(data\_no\_outliers):

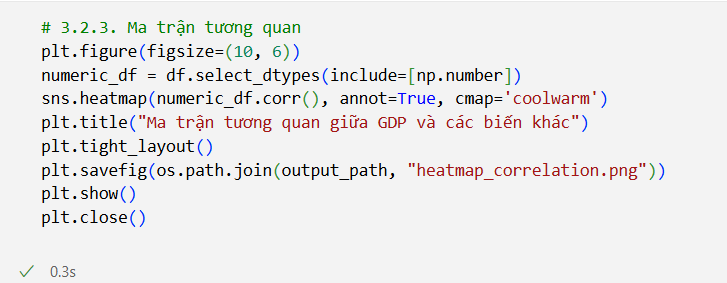
* + Hàm vẽ biểu đồ **Boxplot** thể hiện dữ liệu sau khi loại bỏ ngoại lai.
  + Biểu đồ được **lưu tại thư mục** output/figure với tên *boxplot\_no\_outliers.png*, giúp trực quan hóa kết quả làm sạch dữ liệu.

Kết quả: DataFrame data\_no\_outliers không còn ngoại lai, và boxplot minh họa dữ liệu sạch.



*Hình 3. 12 Boxplot minh họa dữ liệu sạch.*

## Ma trận tương quan

****

*Hình 3. 13 Mã nguồn ma trận tương quan*

**Biểu đồ Heatmap** để thể hiện mối quan hệ tương quan giữa các cột số học trong dữ liệu đã xử lý ngoại lai:

## numeric\_df:

* + Lọc ra các cột có kiểu dữ liệu số trong DataFrame (df.select\_dtypes(include= [np.number])).
  + Bao gồm các cột như: ['Year', 'Labor\_Agriculture', 'Labor\_Industry', 'Labor\_Services', 'Unemployment\_Rate', 'GDP'].

## plt.figure(figsize=(10, 6)):

* + Thiết lập kích thước biểu đồ là 10x6 inch.

## sns.heatmap():

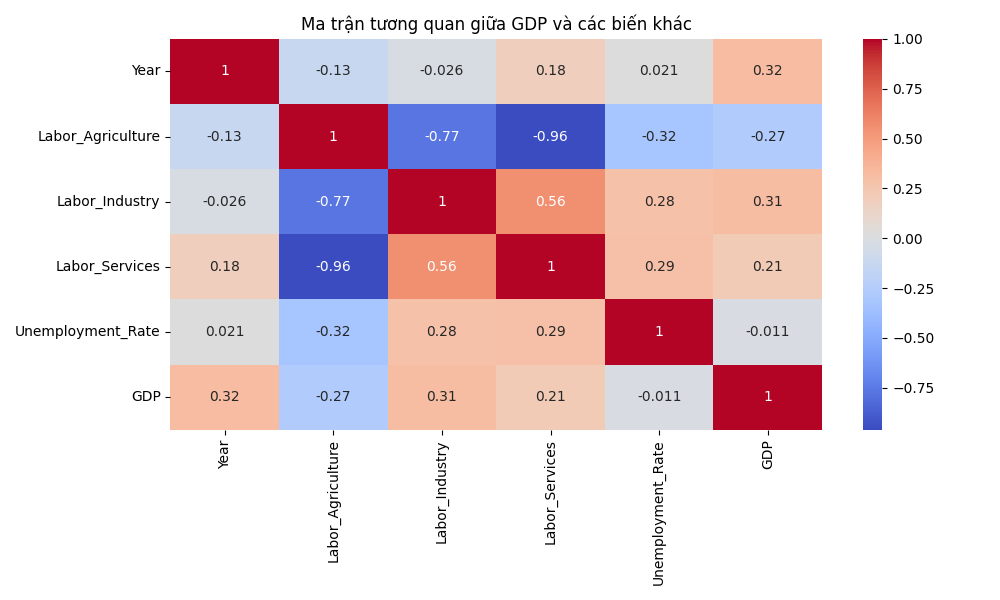
* + Tính toán ma trận tương quan giữa các biến số trong dữ liệu.
  + annot=True: Hiển thị giá trị tương quan trên heatmap.
  + Cmap = ‘coolwarm’:Áp dụng bảng màu giúp dễ nhận biết giá trị tương quan dương(đỏ) và âm (xanh).

## plt.title():

* + Đặt tiêu đề: " Ma trận tương quan giữa GDP và các biến khác”".

1. **plt.savefig():** Lư biểu đồ vào thư mục output/figure với tên heatmap\_correlation.png
2. **plt.show()**: Hiển thị biểu đồ.

**Kết quả**: Heatmap minh họa mức độ tương quan giữa các yếu tố, các vùng màu thể hiện mối quan hệ tuyến tính giữa GDP và các yếu tô kinh tế.



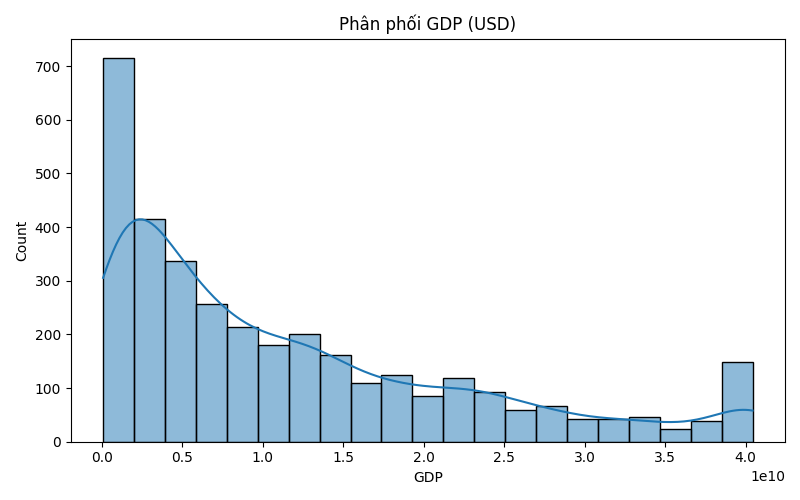
*Hình 3. 14 Kết quả Heatmap minh họa mức độ tương quan giữa các yếu tố*

## Mã nguồn chức năng Trực quan hoá dữ liệu

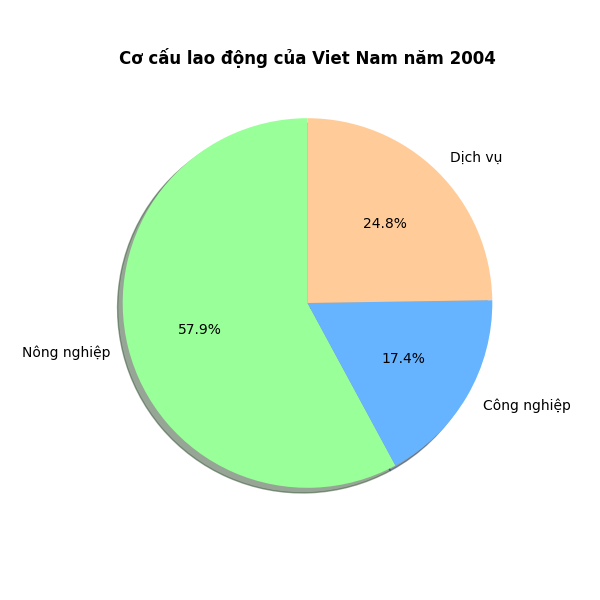
## Biểu đồ thể hiện phân phối đặc trưng các phân loại

****

*Hình 3. 15 Mã nguồn biểu đồ thể hiện phân phối đặc trưng các phân loại*

**

*Hình 3. 16 Kết quả biểu đồ phân phôi GDP*

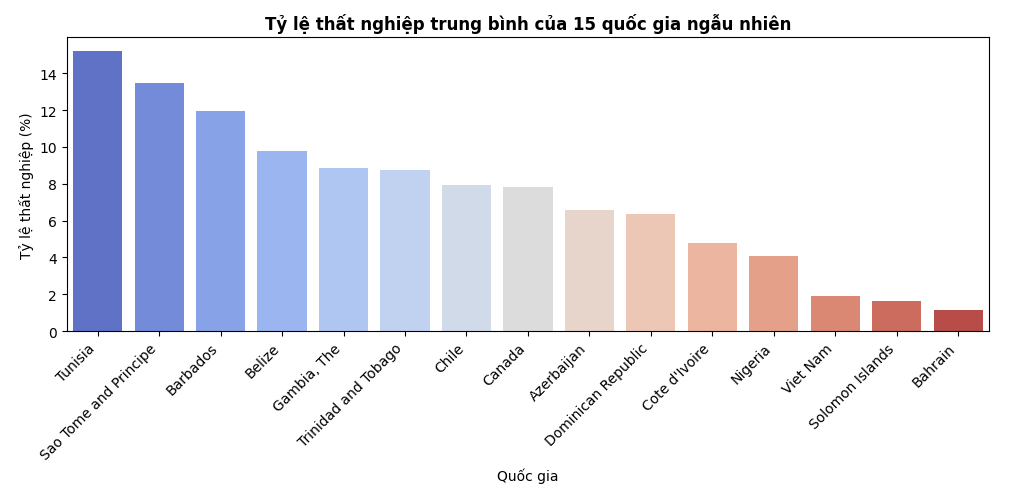


*Hình 3. 17 Kết quả biểu đồ pie thể hiện phân phối đặc trưng các phân loại lao động của Việt Nam 2004*

## Biểu đồ thể hiện tỷ lệ thất nghiệp trung bình của 15 quốc gia ngẫu nhiên

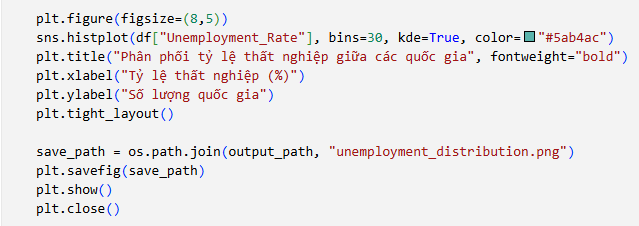
****

*Hình 3. 18 Mã nguồn biểu đồ thể hiện tỷ lệ thất nghiệp trung bình của 15 quốc gia ngâu nhiên*

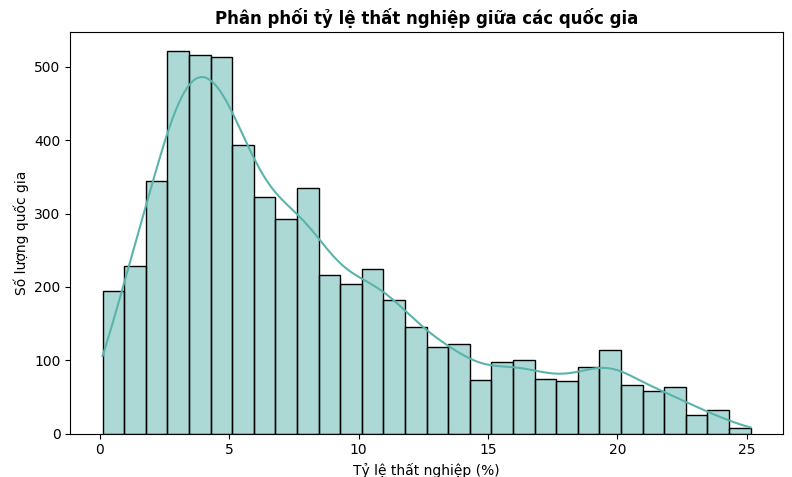


*Hình 3. 19 Kết quả biểu đồ thể hiện tỷ lệ thất nghiệp*

## Biểu đồ thể hiện các đặc trưng liên tục

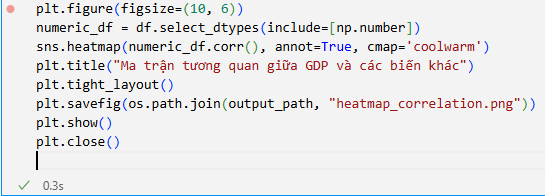
****

*Hình 3. 20 Mã nguồn biểu đồ thể hiện các đặc trưng liên tục*

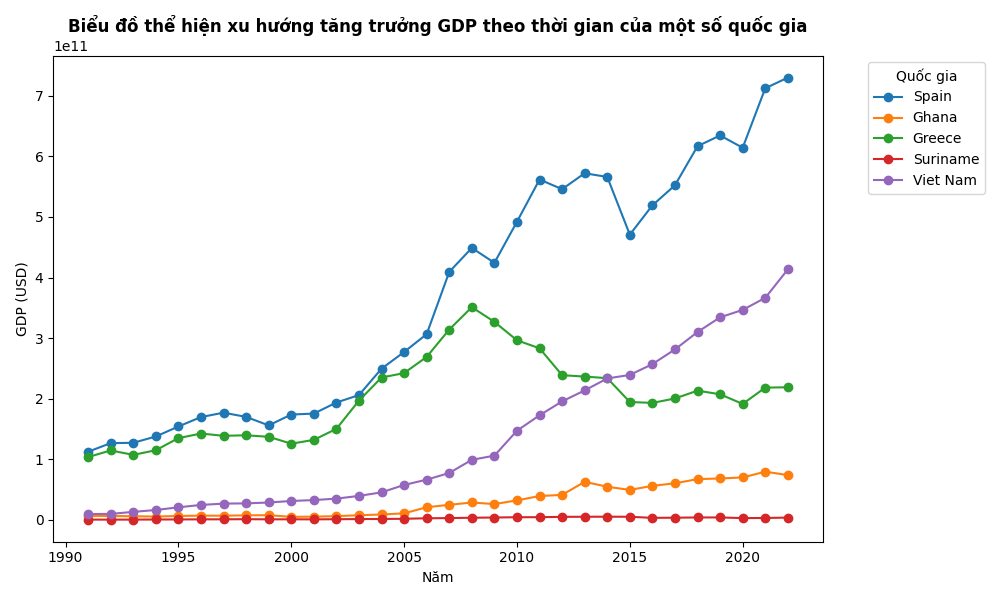
**

*Hình 3. 21 Kết quả biểu đồ thể hiện các đặc trưng liên tục*

## Biểu đồ thể hiện xu hướng tăng GDP theo thời gian của một số quốc gia

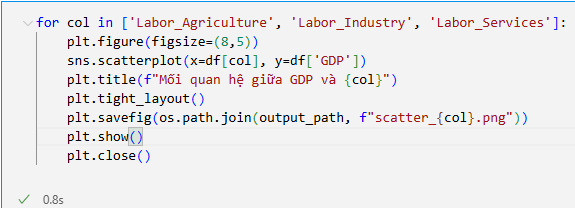
****

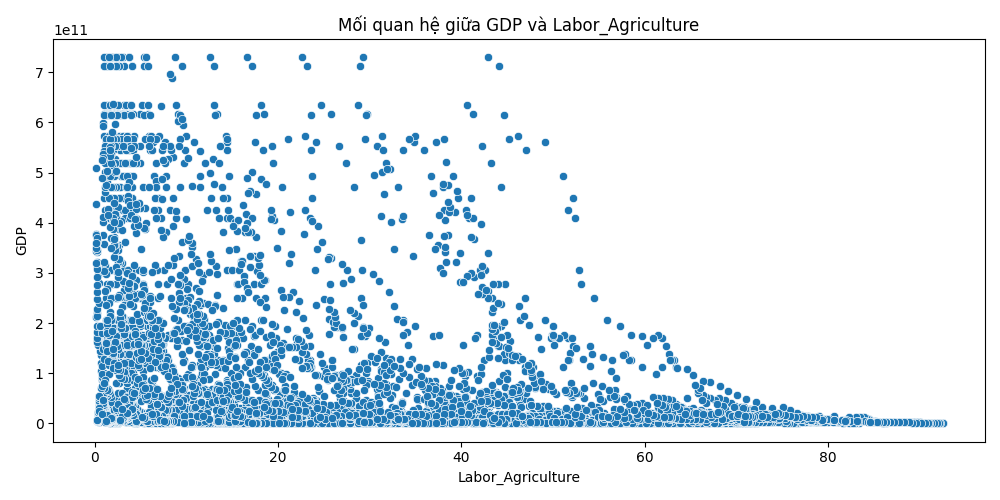
*Hình 3. 22 Mã nguồn biểu đồ thể hiện xu hương tăng GDP theo thời gian của một số quốc gia*

**

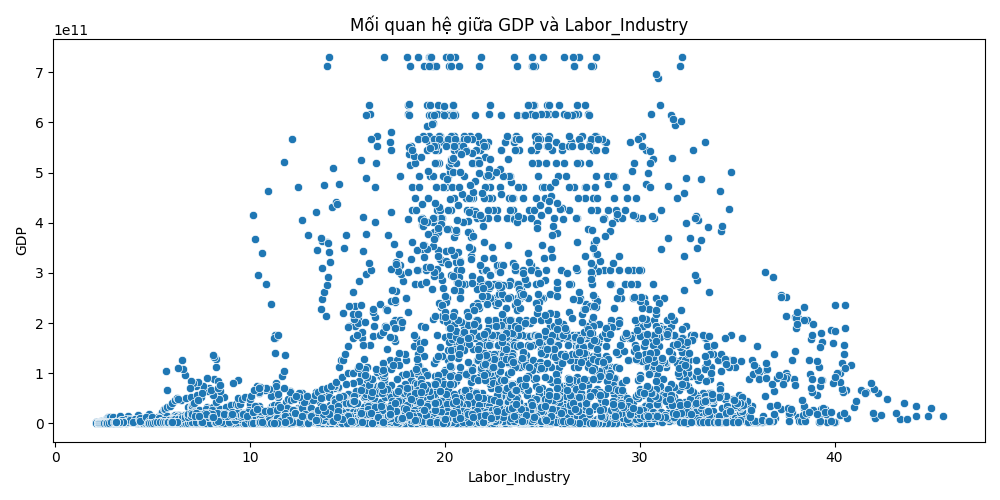
*Hình 3. 23 Kết quả biểu đồ thể hiện xu hương tăng GDP theo thời gian của một số quốc gia*

## Biểu đồ thể hiện Biểu đồ thể hiện mối quan hệ giữa GDP và các nhóm lao động

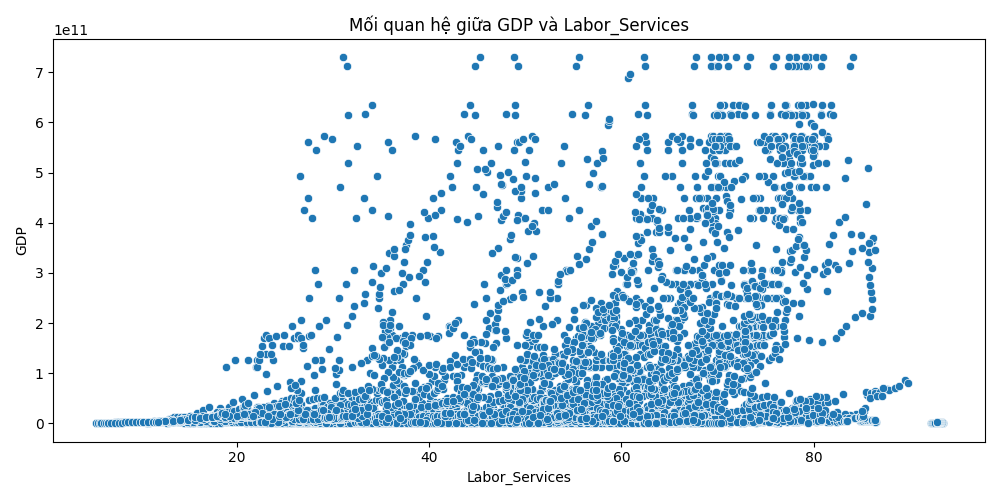
****

*Hình 3. 24 Mã nguồn biểu đồ thể hiện mối tương quan giữa GDP và các phần tử còn lại*

*Hình 3. 25 Kết quả biểu đồ thể hiện mối tương quan giữa GDP và Nông nghiệp*

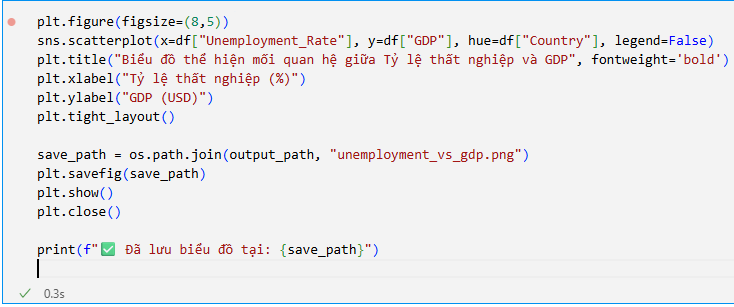
**

*Hình 3. 26 Kết quả biểu đồ thể hiện mối tương quan giữa GDP và Công nghiệp*

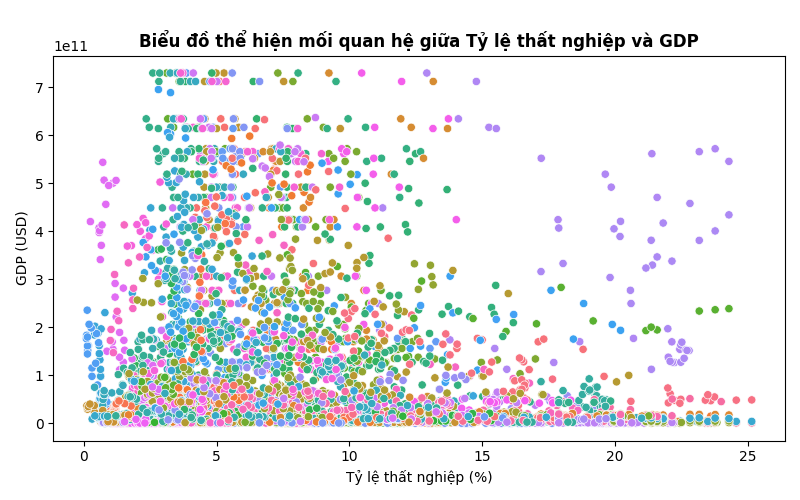
**

*Hình 3. 27 Kết quả biểu đồ thể hiện mối tương quan giữa GDP và Dịch vụ*

## Biểu đồ so sánh GDP thực tế và GDP dự đoán từ mô hình Extra Trees Regressor

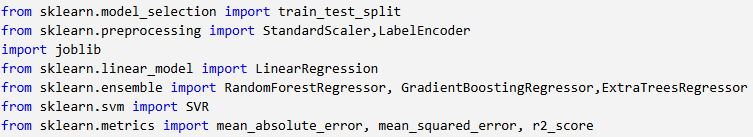
****

*Hình 3. 28 Mã nguồn biểu đồ thể hiện sự trực quan hoá giá trị thiếu*

**

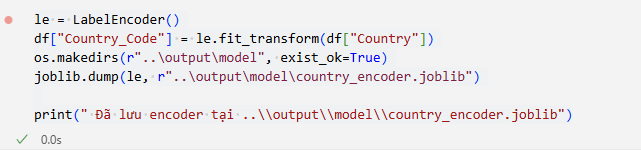
*Hình 3. 29 Kết quả biểu đồ thể hiện sự trực quan hoá giá trị thiếu*

## Chuyển đổi dữ liệu

****

*Hình 3. 30 Import 1 số thư viện cần thiết*

## Xử lý dữ liệu chuỗi

****

*Hình 3. 31 Mã nguồn xử lý dữ liệu chuỗi*

**

*Hình 3. 32 Kết quả xử lý dữ liệu chuỗi*

## Xử lý dữ liệu số

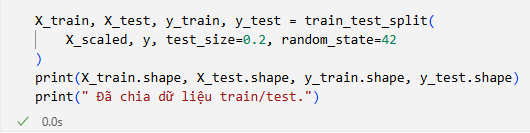


*Hình 3. 33 Mã nguồn xử lý dữ liệu số*

**

*Hình 3. 34 Kết quả xử lý dữ liệu số*

## Chia dữ liệu huấn luyện và kiểm thử

**

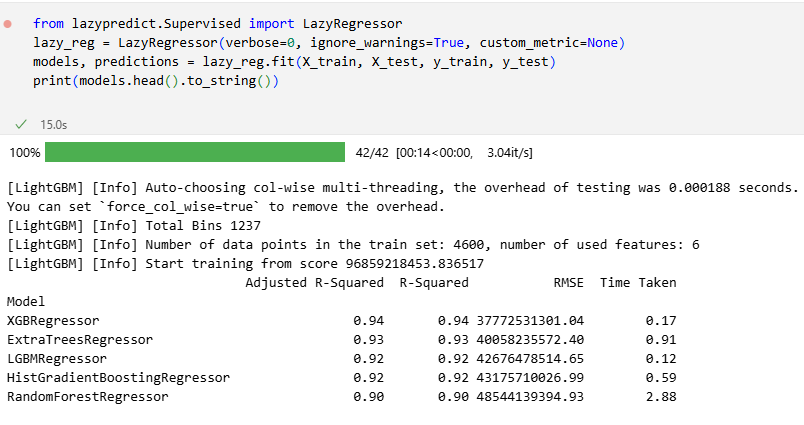
*Hình 3. 35 Mã nguồn chia dữ liệu huấn luyện và kiểm thử*

**

*Hình 3. 36 Mã nguồn chia dữ liệu huấn luyện và kiểm thử*

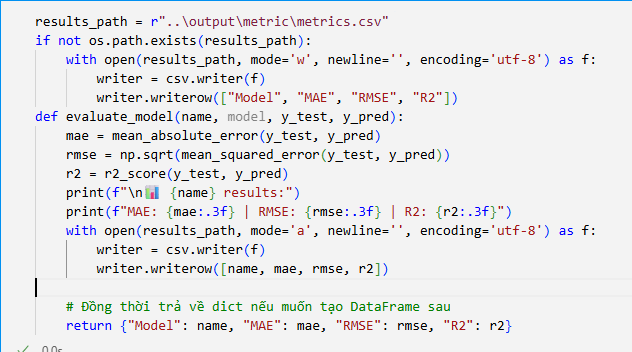
## Huấn luyện mô hình SKLEARN

## Tìm kiếm mô hình

****

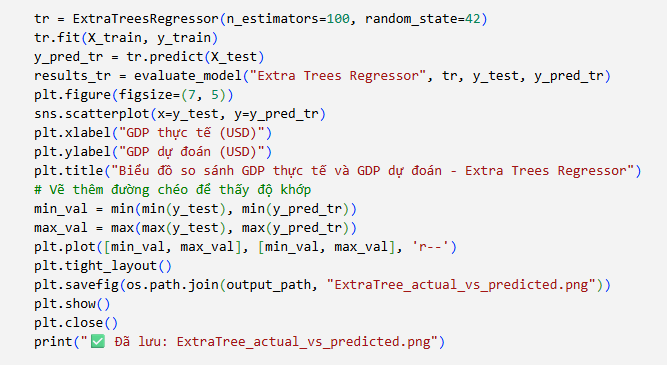
*Hình 3. 37 Mã nguồn tìm kiếm mô hình dùng Lazypredict*

## Đánh giá hiệu suất mô hình

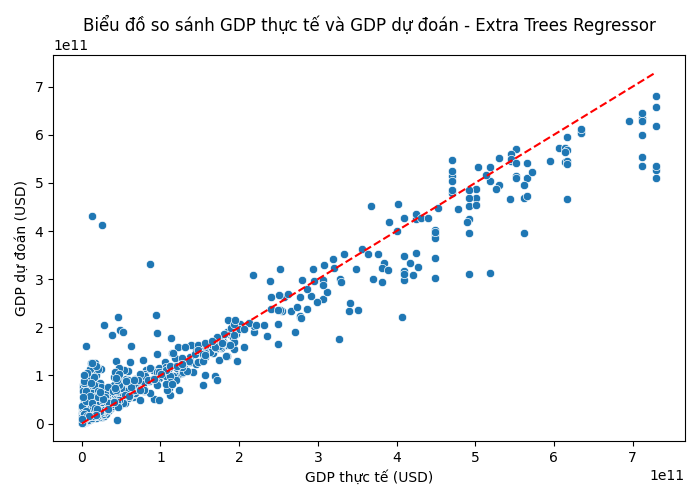


*Hình 3. 38 Mã nguồn đánh giá hiệu suất mô hình*

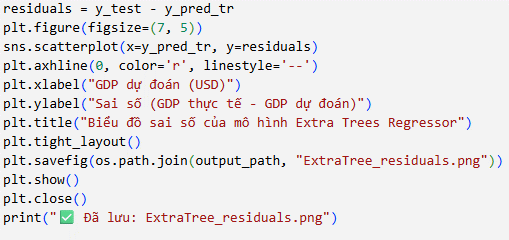
## Huấn luyện mô hình Extra Trees Regressor



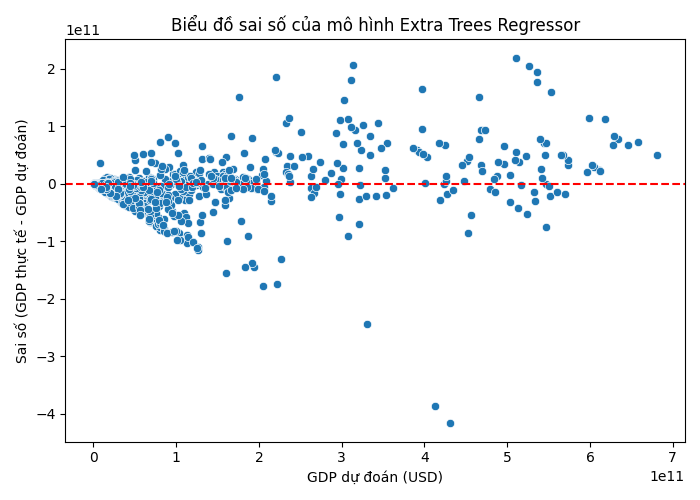
*Hình 3. 37 Mã nguồn so sánh GDP thực tế và GDP dự đoán*



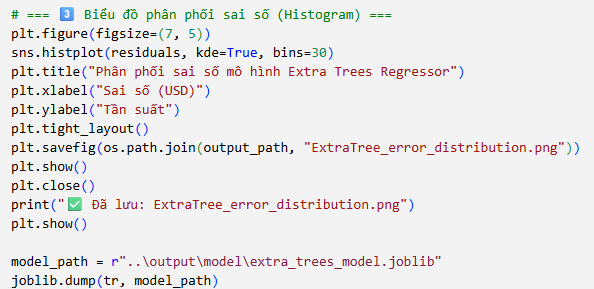
*Hình 3. 38 Kết quả so sánh GDP thực tế và GDP dự đoán*



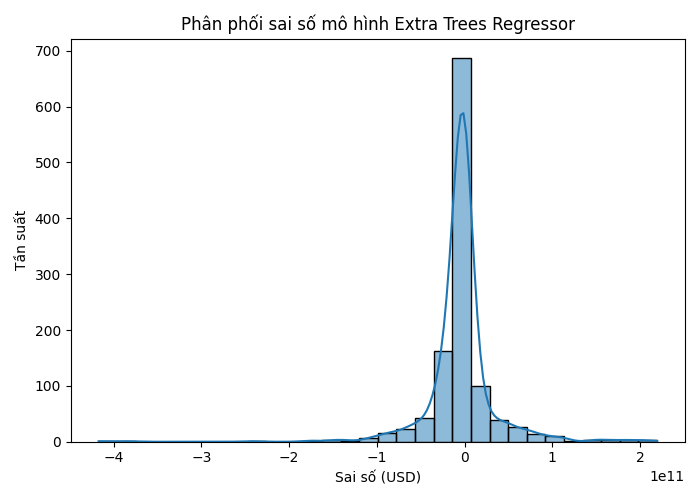
*Hình 3. 39 Mã nguồn sai số GĐP*



*Hình 3. 39 Kết quả sai số GĐP*

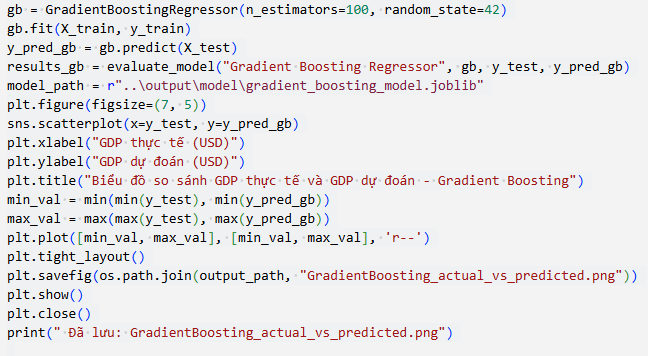


*Hình 3. 310 Mã nguồn tần số phân phối sai số GĐP*

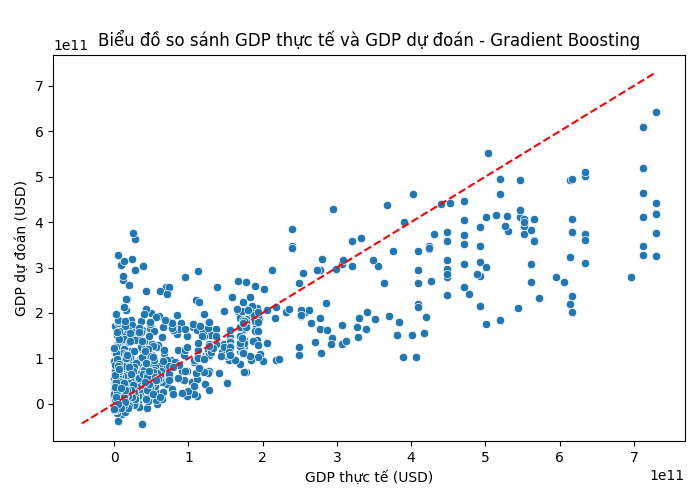


*Hình 3. 311 Kết quả tần số phân phối sai số GĐP*

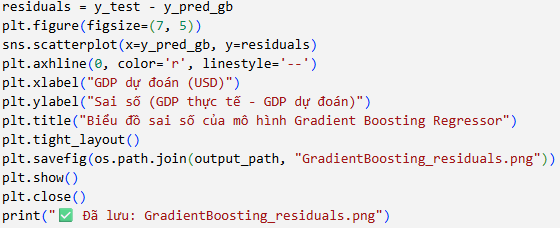
## Huấn luyện mô hình Gradient Boosting Regressor



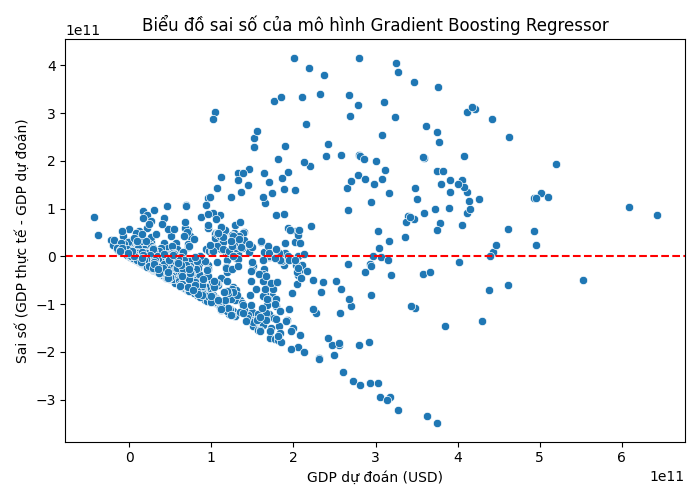
*Hình 3. 312 Mã nguồn biểu đồ so sánh GDP thực tế và GDP dự đoán*



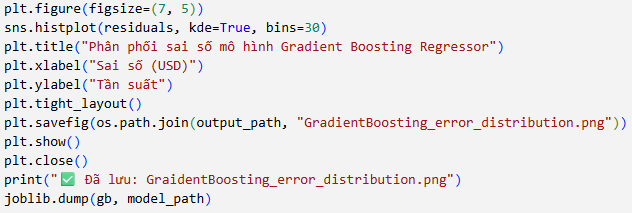
*Hình 3. 313 Kết quả biểu đồ so sánh GDP thực tế và GDP dự đoán*



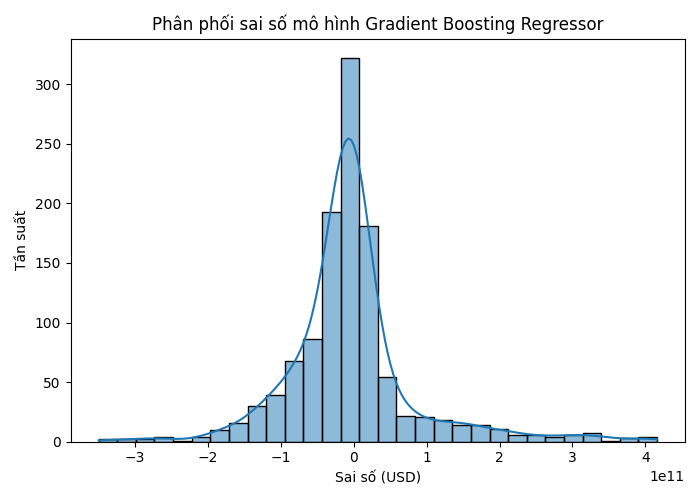
*Hình 3. 314 Mã nguồn biểu đồ sai số GDP*



*Hình 3. 315 Kết quả biểu đồ sai số GDP*

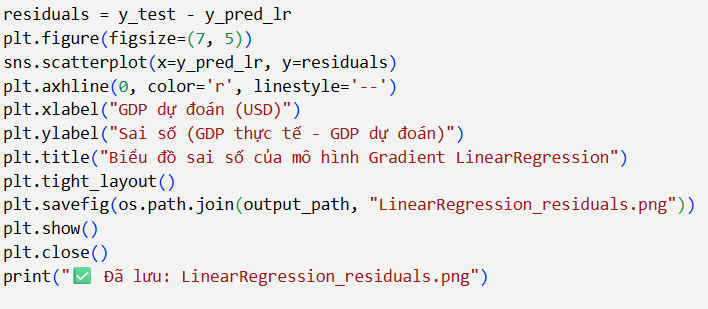


*Hình 3. 316 Mã nguồn biểu đồ phân phối sai số GDP*

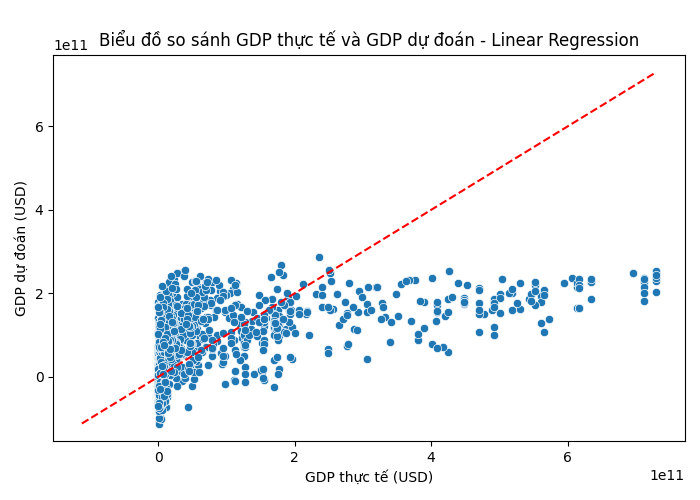


*Hình 3. 317 Kết quả biểu đồ phân phối sai số GDP*

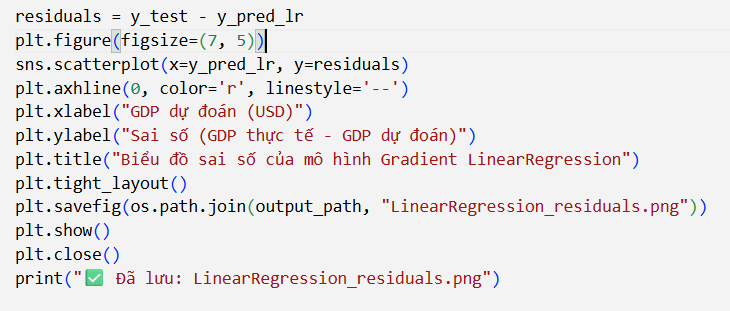
## Huấn luyện mô hình Linear Regression



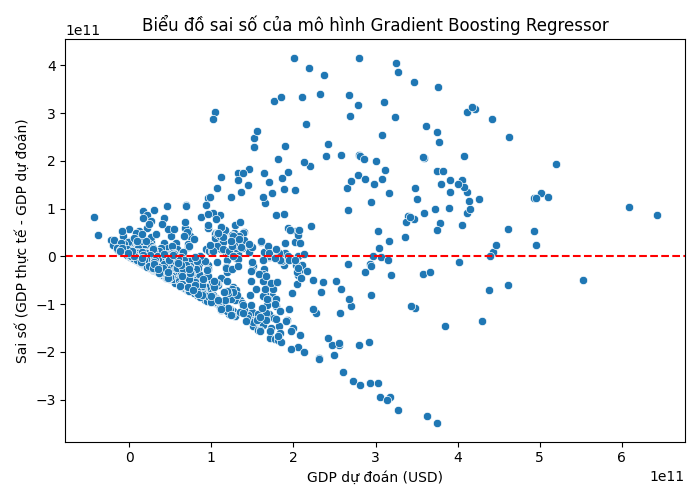
*Hình 3. 318 Mã nguồn biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



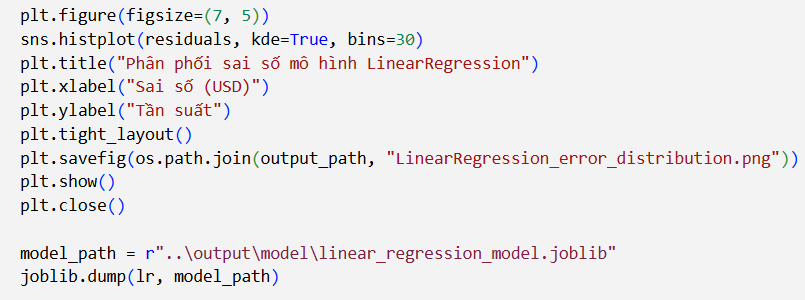
*Hình 3. 319 Kết quả biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



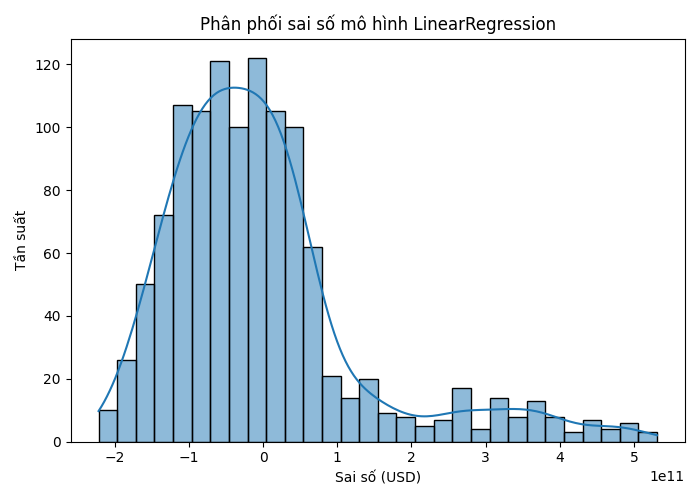
*Hình 3. 320 Mã nguồn biểu đồ sai số GDP*



*Hình 3. 321 Kết quả biểu đồ sai số GDP*

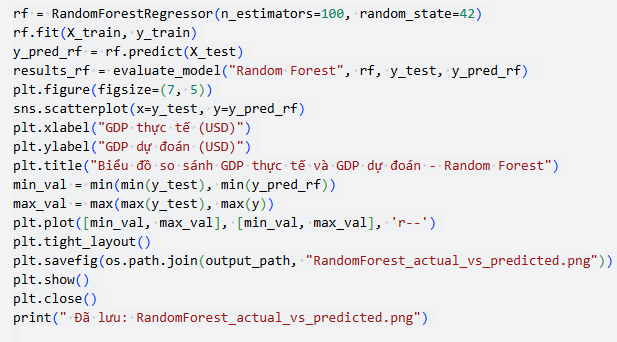


*Hình 3. 322 Mã nguồn phân phối sai số GDP*

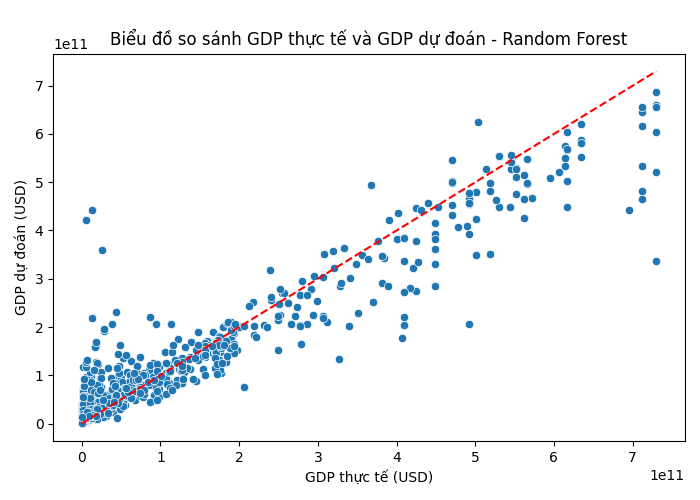


*Hình 3. 323 Kết quả phân phối sai số GDP*

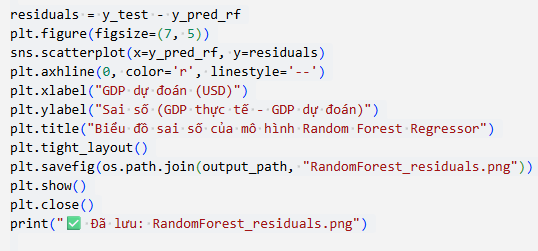
## Huấn luyện mô hình Random Forest Regressor



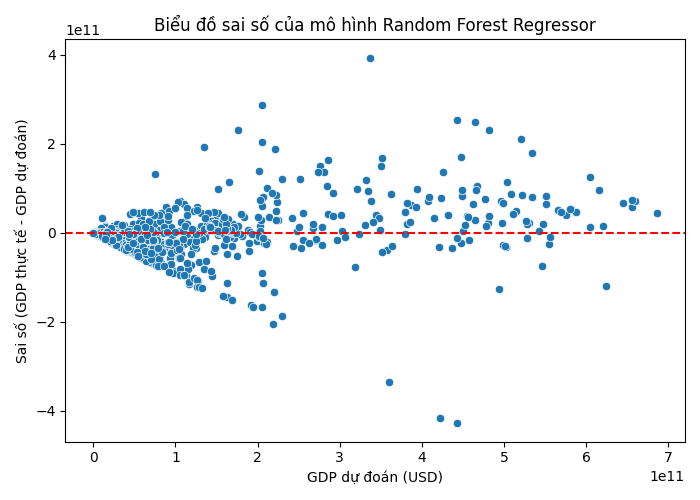
*Hình 3. 324 Mã nguồn biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



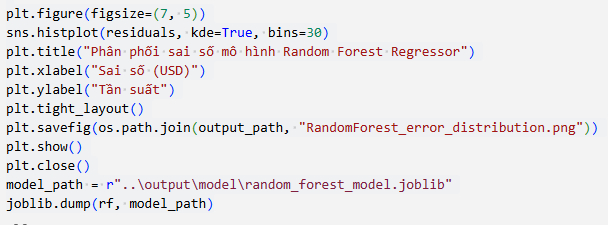
*Hình 3. 325 Kết quả biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



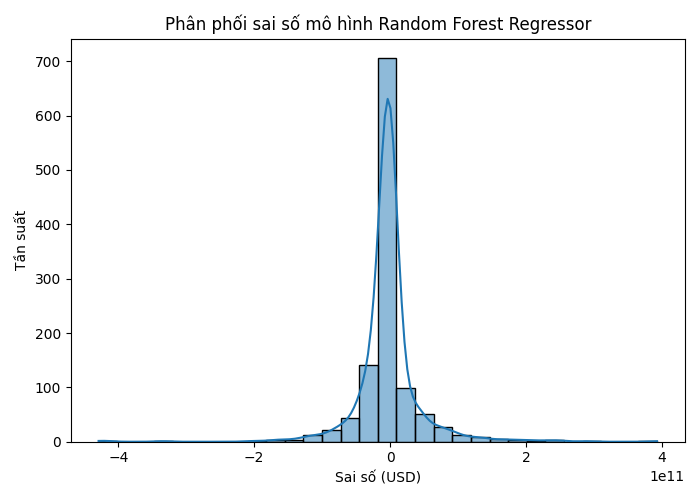
*Hình 3. 326 Mã nguồn biểu đồ sai số GDP*



*Hình 3. 327 Kết quả biểu đồ sai số GDP*

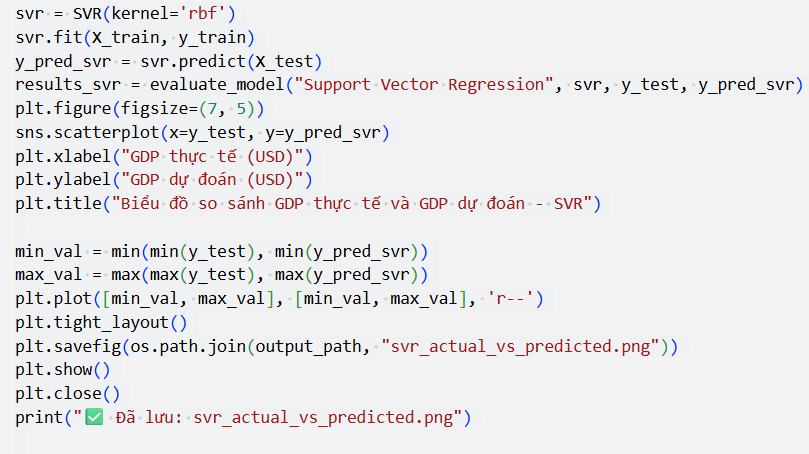


*Hình 3. 328 Mã nguồn phân phối sai số GDP*

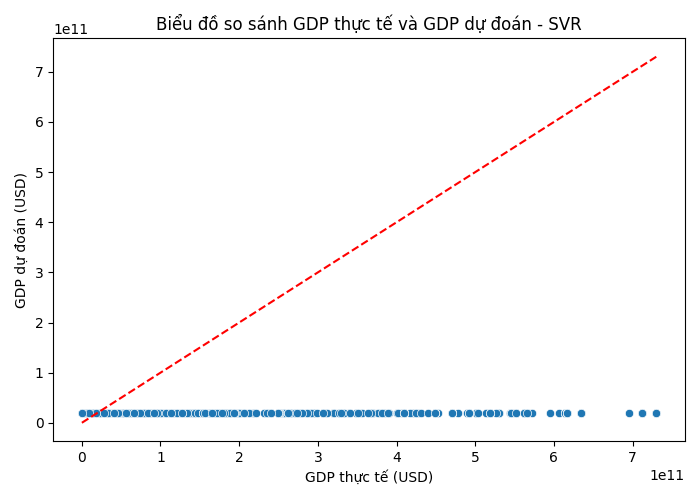


*Hình 3. 329 Kết quả phân phối sai số GDP*

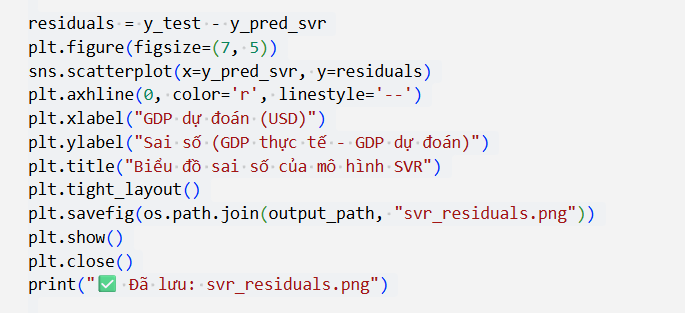
## Huấn luyện mô hình Support Vector Regression (SVR)

x

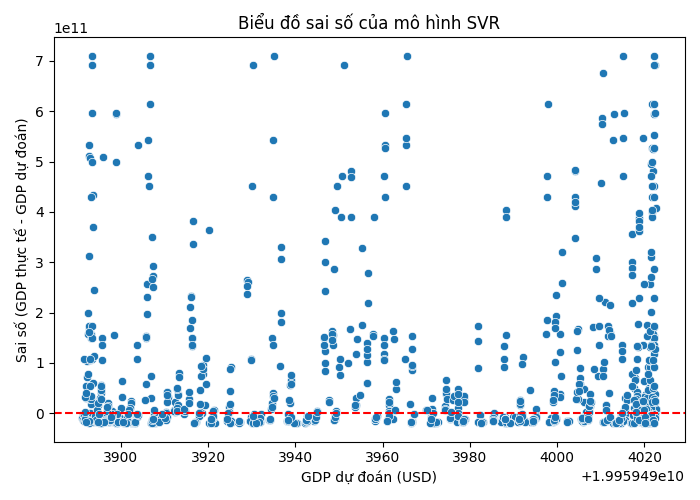
*Hình 3. 330 Mã nguồn biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



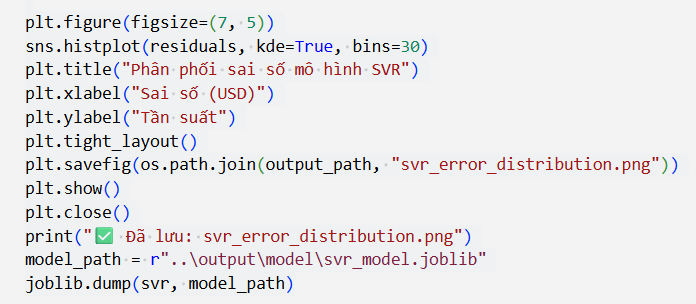
*Hình 3. 331 Kết quả biểu đồ so sánh GDP thực tế vơi GDP dự đoán*



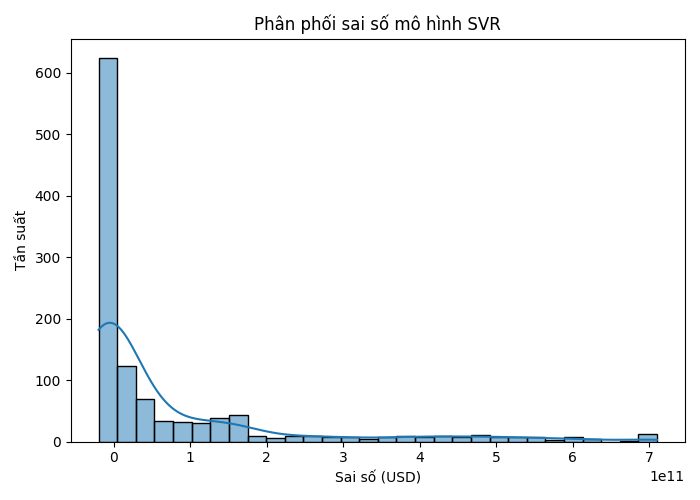
*Hình 3. 332 Mã nguồn biểu đồ sai số GDP*



*Hình 3. 333 Kết quả biểu đồ sai số GDP*

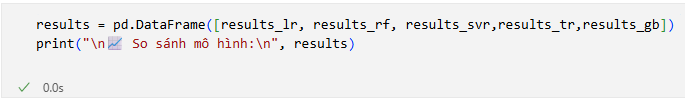


*Hình 3. 334 Mã nguồn phân phối sai số GDP*

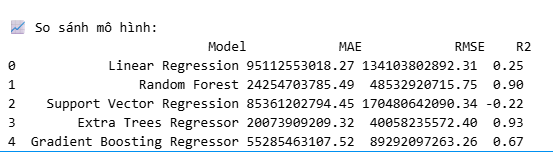


*Hình 3. 335 Kết quả phân phối sai số GDP*

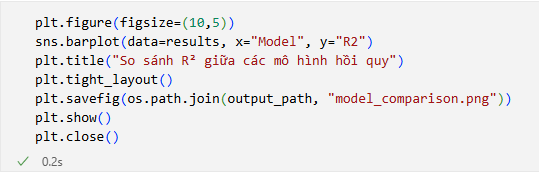
## So sánh và tổng kết giai đoạn huấn luyện



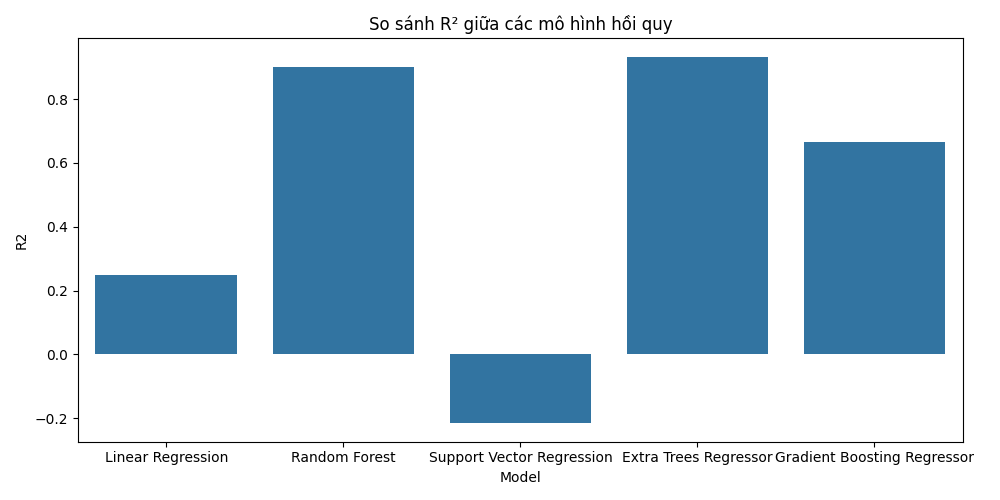
*Hình 3. 336 Mã nguồn so sách các mô hình*

**

*Hình 3. 337 Kết quả so sách các mô hình*

**

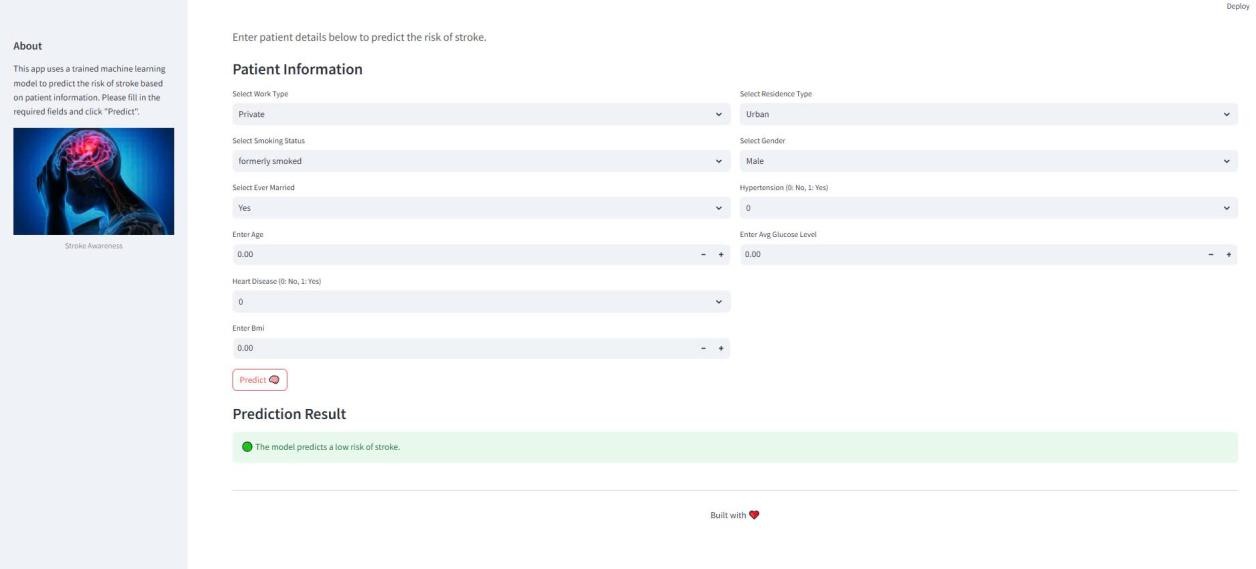
*Hình 3. 337 Mã nguồn so sánh R² giữa các mô hình*

**

*Hình 3. 337 Kết quả so sánh R² giữa các mô hình*

# CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG

Bằng cách sử dụng Streamlit của Python em đã đưa bài toán dự đoán bệnh đột quỵ lên giao diện web.



*HÌnh 4. 1 Sử dụng Streamlit để đưa bài toán lên giao diện web*

# KẾT LUẬN

## Kết quả đạt được trong kiến thức:

* + **Hiểu cách hoạt động và tầm quan trọng của các mô hình RandomForestClassifier, KNeighborsClassifier, và Support Vector Classification:**
    - Hiểu cách RandomForestClassifier tổ hợp nhiều cây quyết định để tăng độ chính xác và khả năng tổng quát hóa.
    - Nắm rõ nguyên tắc hoạt động của KNeighborsClassifier dựa trên người láng giềng gần nhất.
    - Hiểu biết về Support Vector Classification và cách nó tìm siêu phân cách tốt nhất trong không gian cao chiều.

## Áp dụng các mô hình này vào bài toán dự đoán bệnh đột quỵ:

* + - Triển khai RandomForestClassifier, KNeighborsClassifier, và Support Vector Classification để phân loại nguy cơ bị đột quỵ dựa trên các đặc trưng như huyết áp, mỡ máu, tiểu sử bệnh tật.
    - Lựa chọn và tinh chỉnh các tham số quan trọng như số cây, k trong KNN, hay tham số kernel trong SVC để tối ưu hóa hiệu suất mô hình.

## Xử lý, phân tích và trực quan hóa dữ liệu phù hợp với các mô hình:

* + - Thực hiện làm sạch dữ liệu, xử lý các giá trị thiếu, chuẩn hóa dữ liệu để đảm bảo tính nhất quán.
    - Trực quan hóa phân phối nguyên nhân để giúp nhân viên y tế hiểu tầm quan trọng của từng yếu tố.

## Mô hình dự đoán bệnh đột quỵ có thể áp dụng trong hệ thống y tế:

* + - Các mô hình này có thể tích hợp vào hệ thống cảnh báo sớm trong bệnh viện hoặc các trung tâm y tế.
    - Cung cấp khả năng phân tích nhanh theo thời gian thực, tăng cơ hộ chẩn đoán đột quỵ kịp thời.

## Hạn chế của đề tài:

* + **Biến động dữ liệu và yếu tố ngoài:**
    - Biến động nhanh chóng trong yếu tố nguy cơ như huyết áp hay cấp độ oxy có thể ảnh hưởng tới độ chính xác.

## Cân bằng dữ liệu:

* + - Sự mất cân đối trong tập dữ liệu (các trường hợp bị đột quỵ thường rất thấp so với không bị) làm cho mô hình thiên về phân lớp chiếm đa số.

## Giới hạn về tính tối ưu:

* + - Các mô hình như RandomForestClassifier và SVC phụ thuộc nhiều vào tham số tôi ưu hóa, trong khi KNN yêu cầu sự cân nhắc cấn thận với k.

## Hướng phát triển của đề tài:

* + **Tích hợp vào hệ thống y tế:**
    - Phát triển các ứng dụng trên nền tảng web và di động giúp bác sĩ và bệnh nhân giắm thiểu thời gian chẩn đoán.

## Nâng cao tính giải thích:

* + - Sử dụng các kỹ thuật trực quan hóa và giải thích như LIME hoặc SHAP để tăng sự minh bạch và đáng tin cậy của mô hình.

## Mở rộng tập dữ liệu:

* + - Thu thập thêm dữ liệu từ nhiều khu vực và nhóm tuổi để cải thiện tính tổng quát hóa.

# TÀI LIỆU THAM KHẢO

1. Dữ liệu

[Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)

1. Tài liệu

|  |  |
| --- | --- |
| [1] | Giáo trình Học máy cơ bản ( Nhà xuất bản khoa học và kỹ thuật) |
| [2] | Các slide bài thuyết trình Học máy cơ bản – các nhóm trình bày. |
| [3] | Các slide bài giảng Học máy cơ bản – thầy TS. Hoàng Quốc Việt |
| [4] | Deep Learning - Tác giả: Ian Goodfellow, Yoshua Bengio, Aaron Courville |
| [5] | Python Machine Learning - Tác giả: Sebastian Raschka, Vahid Mirjalili |
| [6] | [https://www.coursera.org/learn/machine-learning](http://www.coursera.org/learn/machine-learning) |
| [7] | https://scikit-learn.org/stable/index.html |
| [8] | [https://www.kaggle.com/](http://www.kaggle.com/) |
| [9] | https://chatgpt.com/ |