# **I.MỤC LỤC**

Mục lục

[**I.MỤC LỤC** 1](#_Toc198908671)

[**II.LỜI CẢM ƠN** 3](#_Toc198908672)

[**II.NỘI DUNG BÀI TẬP LỚN** 4](#_Toc198908673)

[**Chương 1.Giới thiệu đầu bài** 4](#_Toc198908674)

[***1.1.Thông tin đề tài*** 4](#_Toc198908675)

[***1.2.Thông tin đề tài*** 4](#_Toc198908676)

[**Chương 2.Cơ sở lý thuyết** 5](#_Toc198908677)

[***2.1.Giới thiệu thư viện pandas*** 5](#_Toc198908678)

[***2.2.Giới thiệu mô hình ARIMA*** 5](#_Toc198908679)

[***2.3.Giới thiệu thư viện Matplotlib*** 6](#_Toc198908680)

[**Chương 3.Thiết kế và xây dựng chương trình** 7](#_Toc198908681)

[***3.1.Sơ đồ khối hệ thống*** 7](#_Toc198908682)

[Hình 1.Sơ đồ phân cấp chức năng 7](#_Toc198908683)

[***3.2.Sơ đồ khối thuật toán chính*** 8](#_Toc198908684)

[***3.2.1.Sơ đồ khối*** 8](#_Toc198908685)

[Hình 2.Sơ đồ khối 8](#_Toc198908686)

[***3.2.2.Phân tích các khối*** 8](#_Toc198908687)

[***3.3.Cấu trúc dữ liệu*** 11](#_Toc198908688)

[***3.1.1.Cấu trúc dữ liệu đầu vào*** 11](#_Toc198908689)

[***3.1.2.Cấu trúc dữ liệu sau tiền xử lý*** 12](#_Toc198908690)

[***3.1.2.Cấu trúc dữ liệu cho mô hình*** 12](#_Toc198908691)

[***3.1.3.Cấu trúc dữ liệu đầu ra*** 12](#_Toc198908692)

[***3.4.Chương trình*** 13](#_Toc198908693)

[***3.4.1.Quá trình huấn luyện*** 13](#_Toc198908694)

[***3.4.2.Cấu trúc chương trình chính*** 14](#_Toc198908695)

[**Chương 4.Thực nghiệm và kết luận** 16](#_Toc198908696)

[***4.1.Thực nghiệm*** 16](#_Toc198908697)

[***4.2.Kết luận*** 16](#_Toc198908698)

[**III.TÀI LIỆU THAM KHẢO** 18](#_Toc198908699)

# **II.LỜI CẢM ƠN**

Trước hết, em xin gửi lời cảm ơn chân thành đến thầy **Nguyễn Văn Huy**, người đã tận tình hướng dẫn, chỉ bảo và hỗ trợ em trong suốt quá trình thực hiện đồ án môn học này. Những ý kiến đóng góp quý báu cùng sự định hướng của thầy đã giúp em hoàn thiện sản phẩm một cách tốt nhất

Em cũng xin cảm ơn các thầy cô trong Khoa Điện Tử, Bộ môn Công Nghệ Thông Tin đã tạo điều kiện thuận lợi để em có cơ hội học tập, nghiên cứu và thực hành, giúp em tích lũy thêm nhiều kiến thức và kinh nghiệm quan trọng trong lĩnh vực tin học công nghiệp

Cuối cùng, em xin gửi lời cảm ơn sâu sắc đến gia đình và bạn bè, những người đã luôn động viên, khích lệ và ủng hộ em trong suốt quá trình thực hiện đồ án. Chính sự hỗ trợ và tin tưởng này đã giúp em vượt qua những khó khăn và hoàn thành tốt đề tài

Em hy vọng sản phẩm này sẽ thể hiện được những nỗ lực và tâm huyết của bản thân, đồng thời góp phần vào việc ứng dụng công nghệ trong lĩnh vực tin học công nghiệp

Thái Nguyên, ngày….tháng…..năm 2025

**SINH VIÊN**

*(Ký ghi rõ họ tên)*

# **II.NỘI DUNG BÀI TẬP LỚN**

## **Chương 1.Giới thiệu đầu bài**

## ***1.1.Thông tin đề tài***

-Tên đề tài: Phân tích dự báo dân số

## ***1.2.Thông tin đề tài***

-Yêu cầu: Xây dựng ứng dụng web phân tích và dự báo dân số theo vùng địa lý

-Dữ liệu đầu vào: Dữ liệu dân số từ World Population Data

[***https://www.kaggle.com/datasets/iamsouravbanerjee/world-population-dataset***](https://www.kaggle.com/datasets/iamsouravbanerjee/world-population-dataset)

Đây là một bộ dữ liệu trên Kaggle về dân số thế giới theo từng quốc gia hoặc vùng lãnh thổ. Bộ dữ liệu này chứa thông tin về dân số từ các năm trước, giúp phân tích xu hướng tăng trưởng dân số, mật độ dân số, tỷ lệ sinh, tuổi trung bình, và nhiều chỉ số khác

-Yêu cầu đầu ra: Dự báo dân số và biểu đồ tăng trưởng dân số theo năm

-Các tính năng:

+Dự báo dân số bằng Time Series

+Visualization dân số theo quốc gia

-Phương pháp:

+Sử dụng Pandas để xử lý dữ liệu

+Sử dụng ARIMA hoặc mô hình Linear Regression dự báo dân số

+Trực quan hóa bằng Matplotlib

## **Chương 2.Cơ sở lý thuyết**

## ***2.1.Giới thiệu thư viện pandas***

Pandas là một thư viện mã nguồn mở trong Python, được sử dụng rộng rãi trong phân tích dữ liệu và khoa học dữ liệu. Được phát triển bởi Wes McKinney vào năm 2008, Pandas cung cấp các công cụ mạnh mẽ để thao tác và xử lý dữ liệu một cách hiệu quả

Thư viện này đặc biệt hữu ích trong việc làm sạch dữ liệu, phân tích số liệu và trực quan hóa thông tin. Với hai cấu trúc dữ liệu chính là Series (mảng một chiều) và DataFrame (bảng hai chiều giống như bảng Excel), Pandas cho phép người dùng nhập dữ liệu từ nhiều nguồn khác nhau như CSV, Excel, SQL hay JSON, đồng thời thực hiện các thao tác như lọc, sắp xếp, nhóm và nối dữ liệu

Một điểm nổi bật của Pandas là khả năng xử lý dữ liệu thiếu (NaN) một cách thông minh, giúp người dùng dễ dàng chuẩn bị dữ liệu trước khi phân tích. Ngoài ra, thư viện này tích hợp tốt với các công cụ quan trọng như NumPy, Matplotlib và Scikit-learn, tạo nên một hệ sinh thái mạnh mẽ để hỗ trợ các nhiệm vụ liên quan đến phân tích và học máy. Với tính linh hoạt và hiệu quả cao, Pandas đã trở thành một công cụ không thể thiếu đối với các nhà khoa học dữ liệu, nhà phân tích kinh doanh và các lập trình viên cần xử lý dữ liệu lớn

## ***2.2.Giới thiệu mô hình ARIMA***

ARIMA (AutoRegressive Integrated Moving Average) là một mô hình thống kê được sử dụng để phân tích và dự báo chuỗi thời gian, đặc biệt khi dữ liệu có xu hướng nhưng không có yếu tố mùa vụ

Mô hình này kết hợp ba thành phần chính:

+Tự hồi quy (AR - AutoRegressive): Dự báo giá trị hiện tại dựa trên các giá trị quá khứ

+Tích phân (I - Integrated): Xử lý dữ liệu không dừng bằng cách lấy sai phân

+Trung bình trượt (MA - Moving Average): Dự báo dựa trên sai số của các bước trước đó

ARIMA thường được sử dụng trong các lĩnh vực như tài chính, kinh tế, khí tượng học và dự báo dân số

## ***2.3.Giới thiệu thư viện Matplotlib***

Matplotlib là một thư viện mạnh mẽ trong Python, được sử dụng để trực quan hóa dữ liệu thông qua các biểu đồ và đồ thị. Nó cho phép người dùng tạo ra nhiều loại biểu đồ khác nhau như biểu đồ đường, biểu đồ cột, biểu đồ tròn, biểu đồ phân tán và nhiều loại biểu đồ khác. Matplotlib đặc biệt hữu ích trong phân tích dữ liệu, giúp người dùng dễ dàng nhận diện xu hướng và mẫu dữ liệu

Thư viện này hoạt động tốt với các cấu trúc dữ liệu như DataFrame của Pandas và mảng NumPy, giúp việc vẽ biểu đồ trở nên linh hoạt và hiệu quả. Một trong những module quan trọng nhất của Matplotlib là Pyplot, cung cấp các hàm đơn giản để tạo và tùy chỉnh biểu đồ. Người dùng có thể dễ dàng thêm tiêu đề, nhãn trục, chú thích và điều chỉnh màu sắc của biểu đồ

Matplotlib được sử dụng rộng rãi trong khoa học dữ liệu, học máy và phân tích kinh doanh

## **Chương 3.Thiết kế và xây dựng chương trình**

## ***3.1.Sơ đồ khối hệ thống***

-Ta có thể lập sơ đồ khối với các module cho hệ thống như sau:

+Thu thập dữ liệu: Nhận dữ liệu từ các nguồn như Kaggle, API thống kê dân số, tệp CSV hoặc Excel

+Tiền xử lý dữ liệu: Làm sạch dữ liệu, xử lý giá trị bị thiếu, chuyển đổi định dạng phù hợp

+Phân tích dữ liệu: Thống kê dân số theo vùng địa lý, xu hướng phát triển

+Xây dựng mô hình dự báo: Sử dụng ARIMA, Linear Regression hoặc mô hình học sâu để dự báo dân số tương lai

+Trực quan hóa dữ liệu: Sử dụng Matplotlib hoặc Seaborn để tạo biểu đồ giúp hiểu rõ xu hướng dân số

+Phần mềm trên web: Phần mềm với giao diện người dùng giúp nhập dữ liệu và xem kết quả

A diagram of a company

AI-generated content may be incorrect.

### Hình 1.Sơ đồ phân cấp chức năng

## ***3.2.Sơ đồ khối thuật toán chính***

## ***3.2.1.Sơ đồ khối***

A diagram of a project

AI-generated content may be incorrect.

### Hình 2.Sơ đồ khối

## ***3.2.2.Phân tích các khối***

-**Chạy chương trình**: Kích hoạt quy trình dự báo, khởi tạo biến và cấu hình ban đầu của hệ thống

+Đầu vào: Các tham số cấu hình, các tham số model cơ bản, lệnh chạy

+Đầu ra: Bắt đầu luồng xử lý, chuyển tiếp đến khối thu thập dữ liệu

-**Thu thập dữ liệu**: Kết nối đến các nguồn dữ liệu (ví dụ: Kaggle, API thống kê dân số) để lấy dữ liệu lịch sử dân số

+Đầu vào: Thông tin kết nối, địa chỉ nguồn dữ liệu, thông số API, file .csv

+Đầu ra: Dữ liệu thô (raw data) chứa thông tin dân số theo từng khoảng thời gian

-**Tiền xử lý dữ liệu**: Làm sạch dữ liệu: loại bỏ các giá trị mất, xử lý ngoại lệ. Chuẩn hóa định dạng dữ liệu (ví dụ: chuyển đổi kiểu số, định dạng ngày tháng)

+Đầu vào: Dữ liệu thô từ khối Data Collection

+Đầu ra: Dữ liệu đã được xử lý (cleaned data) để phục vụ cho các bước phân tích tiếp theo

-**Phân tích dữ liệu**: Phân tích đặc tính của chuỗi thời gian: kiểm tra xu hướng, tính dừng, tính mùa vụ. Áp dụng các kiểm định thống kê (ví dụ: kiểm định Augmented Dickey-Fuller) để xác định xem dữ liệu có dừng hay không. Trích xuất các thông số thống kê quan trọng phục vụ việc lựa chọn tham số cho mô hình ARIMA

+Đầu vào: Dữ liệu đã tiền xử lý từ khối Data Preprocessing

+Đầu ra: Các thông số cần thiết (như xu hướng, đặc trưng của dữ liệu) được truyền tới bước huấn luyện mô hình

-**Huấn luyện và kiểm tra mô hình**: Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra. Sử dụng dữ liệu huấn luyện để tối ưu hóa các tham số của mô hình ARIMA (các giá trị của **p**, **d**, **q**) thông qua các tiêu chí như AIC/BIC hoặc phân tích ACF/PACF. Thực hiện đánh giá sơ bộ mô hình (ví dụ: tính toán lỗi RMSE, MAE) trên tập dữ liệu kiểm tra ban đầu

+Đầu vào: Các đặc trưng từ khối Data Analysis và dữ liệu đã xử lý

+Đầu ra: Các tham số tối ưu ban đầu và một phiên bản mô hình ARIMA đầu tiên

-**Xây dựng và huấn luyện mô hình ARIMA**: Xây dựng chính xác mô hình ARIMA dựa trên các tham số đã xác định. Huấn luyện mô hình sử dụng toàn bộ dữ liệu huấn luyện. Cập nhật và tinh chỉnh mô hình nếu cần dựa trên phản hồi từ quá trình kiểm thử

+Đầu vào: Tập dữ liệu huấn luyện và các tham số p, d, q từ khối Model Training & Validation

+Đầu ra: Mô hình ARIMA đã được huấn luyện hoàn chỉnh, sẵn sàng dự báo giá trị tương lai

-**Dự đoán**: Sử dụng mô hình ARIMA đã huấn luyện để dự báo dân số cho các khoảng thời gian trong tương lai. Xử lý đầu ra của mô hình nhằm đưa ra dự báo một cách trực quan và dễ hiểu

+Đầu vào: Mô hình ARIMA đã xây dựng và dữ liệu liên quan tới thời gian cần dự báo

+Đầu ra: Kết quả dự báo (giá trị dân số dự đoán) cho từng khoảng thời gian

-**Kiểm thử và đánh giá mô hình**: So sánh kết quả dự báo với dữ liệu thực tế (từ tập kiểm tra). Tính toán các chỉ số đánh giá hiệu năng của mô hình như RMSE, MAE, MAPE… Xác định mức độ chính xác của mô hình và đề xuất cải tiến nếu cần

+Đầu vào: Dự đoán từ khối **Dự đoán** và dữ liệu thực tế

+Đầu ra: Báo cáo hiệu suất mô hình và thông tin phản hồi cho bước cải thiện

-**Cải thiện hiệu suất mô hình**: Dựa trên kết quả đánh giá của khối Model Testing/Evaluation, tiến hành điều chỉnh tham số hoặc cải tiến quy trình tiền xử lý, huấn luyện. Lặp lại quy trình Model Training & Validation để nâng cao độ chính xác của mô hình

+Đầu vào: Phản hồi đánh giá hiệu suất từ khối Model Testing/Evaluation

+Đầu ra: Mô hình ARIMA được tối ưu lại, đạt hiệu năng cải tiến***3.2.3.Quan hệ đầu vào-ra giữa các khối***

-“Thu thập dữ liệu” xuất ra dữ liệu thô - “Tiền xử lý” xử lý và chuyển thành dữ liệu sạch.

-“Tiền xử lý” xuất ra dữ liệu đã chuẩn hóa - “Phân tích dữ liệu” rút trích đặc tính của dữ liệu.

-“Phân tích dữ liệu” cung cấp các thông số phân tích - “Huấn luyện,xác thực dữ liệu” sử dụng để tối ưu tham số.

-“Huấn luyện,xác thực dữ liệu” xuất ra bộ tham số và phiên bản mô hình ban đầu - “Xây dựng, huấn luyện mô hình ARIMA” xây dựng và huấn luyện mô hình.

-“Xây dựng, huấn luyện mô hình ARIMA” cung cấp mô hình hoàn thiện - “Dự đoán” sử dụng mô hình để dự báo.

-“Dự đoán” tạo ra kết quả dự báo - “Kiểm thử mô hình” so sánh với dữ liệu thực tế để đánh giá.

-“Kiểm thử mô hình” đem lại thông tin hiệu suất - “Cải thiện hiệu suất” để cải tiến và quay lại vòng huấn luyện

## ***3.3.Cấu trúc dữ liệu***

## ***3.1.1.Cấu trúc dữ liệu đầu vào***

Dữ liệu dân số: Được lưu trữ dưới dạng bảng (DataFrame của Pandas), gồm các trường như

+country: tên quốc gia

+year: năm thu thập dữ liệu

+population: dân số tại năm đó

+growth\_rate:tốc độ tang trưởng dân số

Định dạng tệp: Dữ liệu đầu vào có thể ở dạng csv, json hoặc truy xuất từ API

## ***3.1.2.Cấu trúc dữ liệu sau tiền xử lý***

Dữ liệu làm sạch: Sau các bước tiền xử lý như loại bỏ giá trị thiếu, xử lý ngoại lệ và chuẩn hóa định dạng, dữ liệu được lưu lại với định dạng tương tự bảng nhưng đảm bảo tính nhất quán. Bổ sung thêm 1 trường là status

+country: tên quốc gia

+year: năm thu thập dữ liệu

+population: dân số tại năm đó

+growth\_rate: tốc độ tang trưởng dân số

+status: chỉ ra dữ liệu hợp lệ hoặc được hiệu chỉnh (nếu có sự thay đổi sau tiền xử lý)

Định dạng dữ liệu: Chuyển đổi kiểu dữ liệu cho phù hợp: year chuyển thành số nguyên, population thành số nguyên hoặc float, và các giá trị growth\_rate sau khi tính toán có thể được làm tròn với số chữ số thập phân phù hợp

## ***3.1.2.Cấu trúc dữ liệu cho mô hình***

Dữ liệu huấn luyện và kiểm thử: Sau khi tiền xử lý, dữ liệu được phân chia thành:

+Tập huấn luyện (Training Data): chứa dữ liệu lịch sử đầy đủ, được sử dụng để huấn luyện mô hình dự báo

+Tập kiểm thử (Testing/Validation Data): dùng để đánh giá và kiểm định độ chính xác của mô hình

Tham số mô hình: Lưu trữ thông tin về các tham số của mô hình dự báo (ví dụ: các giá trị p, d, q trong ARIMA hoặc hệ số trong mô hình Linear Regression) để tiện cho quá trình huấn luyện và tinh chỉnh

Parameters DataFrame/Dict: lưu các giá trị tham số kèm theo các chỉ số hiệu suất ban đầu như RMSE, MAE

## ***3.1.3.Cấu trúc dữ liệu đầu ra***

Bảng dự báo: Kết quả dự báo được lưu trữ dưới dạng bảng với các trường:

+year: năm dự báo

+predicted\_population: dân số dự báo cho năm tương ứng

+confidence\_interval: các cột upper\_bound và lower\_bound nếu mô hình hỗ trợ khoảng tin cậy

Dữ liệu trực quan hóa: Các bảng dữ liệu kết quả được chuyển đổi hoặc tích hợp với các thư viện vẽ biểu đồ như Matplotlib hoặc Seaborn để hiển thị dưới dạng biểu đồ đường, cột... Dữ liệu này cũng có thể được xuất ra dưới dạng CSV hoặc JSON để phục vụ cho các giao diện web hoặc báo cáo trực quan

## ***3.4.Chương trình***

Trong phần này, chương trình được chia thành nhiều hàm độc lập, mỗi hàm phụ trách một nhiệm vụ cụ thể từ thu thập dữ liệu, tiền xử lý, phân tích, huấn luyện mô hình cho đến dự báo và xuất kết quả. Điều này không chỉ giúp kiểm thử và phát triển riêng lẻ từng module mà còn tạo ra luồng xử lý logic, rõ ràng cho toàn bộ ứng dụng

## ***3.4.1.Quá trình huấn luyện***

- ***Hàm thu thập dữ liệu***

+Kết nối và lấy dữ liệu dân số từ các nguồn như file CSV, JSON hoặc API

+Nhiệm vụ chính: Kiểm tra định dạng, kết nối, xử lý lỗi và trả về dữ liệu dạng DataFrame

***-Hàm tiền xử lý dữ liệu***

+Làm sạch và chuẩn hóa dữ liệu thô nhằm loại bỏ dữ liệu thiếu, xử lý ngoại lệ và định dạng lại dữ liệu cho phù hợp với yêu cầu phân tích

+Nhiệm vụ chính: Tạo ra bộ dữ liệu nhất quán với các trường như country, year, population và growth\_rate được xử lý đúng định dạng

***-Hàm phân tích và trực quan hóa dữ liệu***

+Phân tích các đặc tính của dữ liệu (xu hướng dân số, tốc độ tăng trưởng,...) và tạo ra các biểu đồ trực quan để hỗ trợ quá trình ra quyết định

+Nhiệm vụ chính: Tính toán các chỉ số phân tích dữ liệu và hiển thị kết quả qua đồ thị; giúp phát hiện xu hướng và bất thường

-***Hàm huấn luyện và kiểm thử mô hình***

+Huấn luyện mô hình dự báo (chẳng hạn ARIMA hoặc Linear Regression) dựa vào dữ liệu huấn luyện, kiểm thử mô hình và điều chỉnh tham số theo tiêu chí thống kê như AIC/BIC

+Nhiệm vụ chính: Chia dữ liệu, tối ưu tham số mô hình, đánh giá sơ bộ và trả về mô hình đã được tối ưu hoá

***-Hàm dự báo dân số***

+Sử dụng mô hình đã huấn luyện để dự báo dân số cho các khoảng thời gian tương lai

+Tạo bảng kết quả với năm và dân số dự báo, hỗ trợ kiểm chứng qua khoảng tin cậy nếu mô hình cho phép

- ***Hàm xuất kết quả và lưu trữ***

+Xuất kết quả dự báo và lưu các biểu đồ, báo cáo vào file hoặc hiển thị trên giao diện người dùng

+Nhiệm vụ chính: Lưu trữ các kết quả phân tích dự báo cũng như các hình ảnh trực quan vào file báo cáo hoặc trình bày qua giao diện ứng dụng

***- Hàm huấn luyện và kiểm thử mô hình***

+Ghi lại các lỗi phát sinh trong từng bước của quá trình thực thi chương trình, đảm bảo cho việc kiểm soát và khắc phục vấn đề

+Đảm bảo chương trình không dừng đột ngột và người vận hành có thể theo dõi quá trình xử lý thông qua log

## ***3.4.2.Cấu trúc chương trình chính***

Mỗi hàm được xây dựng theo cấu trúc mô-đun, thông qua đó, việc kiểm thử độc lập cũng như dễ dàng bảo trì, mở rộng được đảm bảo. Kết hợp thêm xử lý lỗi và ghi log giúp toàn bộ hệ thống hoạt động ổn định và minh bạch trong quá trình vận hành

Chương trình chính sẽ thực hiện các bước sau:

-Thu thập dữ liệu

+data\_raw = collect\_data(source, params)

-Tiền xử lý dữ liệu

+data\_clean = preprocess\_data(data\_raw)

-Phân tích và trực quan hóa

+summary, fig = analyze\_and\_visualize(data\_clean)

-Phân chia dữ liệu huấn luyện và kiểm thử

+train\_data, test\_data = split\_data(data\_clean)

-Huấn luyện mô hình

+model, metrics = train\_model(train\_data, model\_params)

-Dự báo dân số

+predictions = forecast\_population(model, data\_clean, periods)

-Xuất kết quả và lưu trữ

+output\_results(predictions,"predictions.csv")

+display\_chart(fig)

## **Chương 4.Thực nghiệm và kết luận**

## ***4.1.Thực nghiệm***

Giao diện chính chương trình

A screenshot of a computer

AI-generated content may be incorrect.

## ***4.2.Kết luận***

-Phần mềm đã đạt được:

+Sản phầm đã thu thập được dữ liệu dân số từ nhiều nguồn thông dụng:json,csv,api, tiền xử lý được giá trị thiếu, làm sạch dữ liệu, chuyển đổi dữ liệu, xử lý ngoại lệ, giúp cho việc phân tích được tốt nhất

+ Qua việc áp dụng các thư viện Python như Pandas và Matplotlib, sản phẩm đã phát triển các biểu đồ trực quan thể hiện xu hướng dân số và tốc độ tăng trưởng qua các năm. Nhờ đó, người dùng có thể nhanh chóng nhận diện các mô hình phát triển dân số theo thời gian

+ Sản phẩm đã tích hợp mô hình ARIMA (hoặc mô hình Linear Regression) để dự báo dân số trong tương lai dựa trên dữ liệu lịch sử. Các tham số của mô hình được tối ưu hóa qua tiêu chí thống kê (AIC/BIC) và đánh giá bằng các chỉ số như RMSE, MAE, giúp nâng cao độ chính xác của dự báo.

+ Với việc tổ chức các hàm theo mô-đun, sản phẩm đã xây dựng được một hệ thống phân tích dự báo dân số có khả năng tích hợp trên giao diện web, cho phép người dùng nhập liệu và xem kết quả dự báo một cách trực quan và tiện lợi

-Kiến thức đã học được qua quá trình làm đề tài:

+Kỹ năng xử lý dữ liệu:pandas

+Quá trình áp dụng và tinh chỉnh mô hình ARIMA đã giúp nâng cao hiểu biết về phân tích chuỗi thời gian, cách chọn lựa và tối ưu các tham số mô hình nhằm đưa ra dự báo chính xác

+Qua khâu tích hợp giao diện người dùng, tạo luồng xử lý rõ ràng và các hàm mô-đun riêng biệt đã giúp học được cách thiết lập một hệ thống từ thu thập dữ liệu đến hiển thị kết quả dự báo, đảm bảo khả năng mở rộng và bảo trì hệ thống

-Hướng cải tiến phần mềm trong tương lai:

+Sẽ cân nhắc bổ sung và so sánh thêm các mô hình dự báo hiện đại, chẳng hạn như các mô hình học sâu (LSTM, GRU) hoặc các phương pháp ensemble, để tối ưu hơn nữa khả năng dự báo dân số

+ Cải thiện cách lưu trữ và truy xuất dữ liệu, áp dụng các phương pháp xử lý dữ liệu lớn (Big Data) khi dữ liệu đầu vào của dự án mở rộng, nhằm đảm bảo hiệu suất xử lý không bị giảm sút

+Tiếp tục phát triển và cải tiến giao diện web, bổ sung các tính năng tương tác như bộ lọc tìm kiếm, hiển thị biểu đồ động, báo cáo chi tiết, cũng như tích hợp đa ngôn ngữ để phục vụ đối tượng người dùng rộng rãi hơn

+Phát triển bộ công cụ kiểm thử tự động và cơ chế ghi log chi tiết hơn để theo dõi và xử lý lỗi nhanh chóng, đảm bảo hệ thống hoạt động ổn định và minh bạch trong quá trình vận hành

# **III.TÀI LIỆU THAM KHẢO**