**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin.**

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

**Sinh viên: . . . . . . . . . . . . .LÊ ĐỨC TOÀN . . . . . . . . . . . . . . . . . .**

**Lớp: . . . . . . . . . . . . . K57KMT. . . . . . . . . . . . . . . . . . . . . . . . . . . .**

**Giáo viên GIẢNG DẠY: . . . . . . . . TS.NGUYỄN VĂN HUY. . . . . .**

**Link GitHub: ..** **https://github.com/DucToan25/baitaplonthayhuy….**

**.**

**THÁI NGUYÊN - 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên: LÊ ĐỨC TOÀN*

*Lớp*: ……K57KMT…. *Ngành:………* KĨ THUẬT MÁY TÍNH…………………………

*Giáo viên hướng dẫn: NGUYỄN VĂN HUY*

*Ngày giao đề*  *Ngày hoàn thành*

*Tên đề tài :  Dự báo điểm của sinh viên*

*Yêu cầu :*

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày….tháng…..năm…..*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên*

Mục Lục

[LỜI NÓI ĐẦU 5](#_Toc199445898)

[LỜI CẢM ƠN 6](#_Toc199445899)

[CHƯƠNG I: KHẢO SÁT HỆ THỐNG 7](#_Toc199445900)

[I. Tổng quan về dự báo điểm thi 7](#_Toc199445901)

[II. Khảo sát các mô hình học máy 7](#_Toc199445902)

[III. Khảo sát yêu cầu dự báo 8](#_Toc199445903)

[CHƯƠNG II: PHÂN TÍCH HỆ THỐNG 9](#_Toc199445904)

[I. Phân tích yêu cầu hệ thống 9](#_Toc199445905)

[II. Phân tích kiến trúc hệ thống 9](#_Toc199445906)

[CHƯƠNG III: THIẾT KẾ HỆ THỐNG CÀI ĐẶT CHƯƠNG TRÌNH 11](#_Toc199445907)

[I. Thiết kế kiến trúc triển khai 11](#_Toc199445908)

[1.1. Lựa chọn phương án triển khai Phương án sử dụng Python và Scikit-learn được chọn vì: 11](#_Toc199445909)

[- Đơn giản: Dễ triển khai trên môi trường local. 11](#_Toc199445910)

[- Hiệu quả: Các thư viện Python hỗ trợ tốt cho học máy. 11](#_Toc199445911)

[- Mở rộng: Dễ thêm mô hình hoặc tích hợp với các hệ thống khác. 11](#_Toc199445912)

[1.2. Thiết kế mô hình triển khai 11](#_Toc199445913)

[- Tải dữ liệu: Sử dụng data\_loader.py để đọc và khám phá dữ liệu. 11](#_Toc199445914)

[- Tiền xử lý: data\_loader.py chuẩn hóa dữ liệu và chia tập train/test. 11](#_Toc199445915)

[- Huấn luyện mô hình: model.py tạo pipeline kết hợp tiền xử lý và mô hình. 11](#_Toc199445916)

[- Trực quan hóa: data\_visualization.py vẽ các biểu đồ phân tích. 11](#_Toc199445917)

[- Lưu trữ: Lưu mô hình và kết quả vào thư mục output. 11](#_Toc199445918)

[II. Quy trình cài đặt hệ thống 12](#_Toc199445919)

[III. Thiết kế mô hình dự bá 13](#_Toc199445920)

[CHƯƠNG IV: NHẬN XÉT VÀ ĐÁNH GIÁ 15](#_Toc199445921)

[I. Kết quả đạt được 15](#_Toc199445922)

[II. Hạn chế và vấn đề gặp phải 15](#_Toc199445923)

[III. Đề xuất cải tiến 16](#_Toc199445924)

[KẾT LUẬN 17](#_Toc199445925)

# 

# **LỜI NÓI ĐẦU**

Trong bối cảnh giáo dục hiện đại, việc dự báo điểm thi của sinh viên là một nhiệm vụ quan trọng nhằm hỗ trợ việc đánh giá hiệu quả học tập, tối ưu hóa quá trình giảng dạy và cung cấp các can thiệp kịp thời cho sinh viên. Sự phát triển của các công nghệ trí tuệ nhân tạo và học máy đã mở ra cơ hội xây dựng các mô hình dự báo chính xác, tận dụng dữ liệu học tập để đưa ra các dự đoán đáng tin cậy.

Báo cáo này được thực hiện trong khuôn khổ môn học Lập trình Python dưới sự hướng dẫn của TS. Nguyễn Tuấn Linh, nhằm nghiên cứu và xây dựng một hệ thống dự báo điểm thi của sinh viên dựa trên ngôn ngữ lập trình Python và các thư viện học máy như Scikit-learn. Dự án không chỉ là bài tập học thuật mà còn là cơ hội để áp dụng kiến thức lập trình vào việc phân tích dữ liệu và xây dựng mô hình dự báo.

Báo cáo trình bày chi tiết quá trình khảo sát dữ liệu, phân tích yêu cầu, thiết kế và triển khai hệ thống dự báo điểm thi, cũng như đánh giá hiệu quả của các mô hình học máy được sử dụng. Chúng tôi cũng chia sẻ các khó khăn gặp phải và các giải pháp khắc phục, cung cấp một tài liệu tham khảo hữu ích cho những ai quan tâm đến việc ứng dụng Python trong phân tích và dự báo dữ liệu giáo dục.

# **LỜI CẢM ƠN**

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc đến TS. Nguyễn Tuấn Linh, giảng viên hướng dẫn, đã tận tình chỉ bảo, cung cấp những định hướng chuyên môn quý báu và hỗ trợ em trong suốt quá trình thực hiện đề tài. Những góp ý và sự động viên của thầy là nguồn động lực lớn giúp em vượt qua các khó khăn để hoàn thành dự án.

Em cũng xin cảm ơn các giảng viên trong Khoa Điện tử, Trường Đại học Kỹ thuật Công nghiệp, đã trang bị cho em nền tảng kiến thức vững chắc về lập trình Python và học máy, tạo điều kiện thuận lợi để thực hiện dự án này.

Đặc biệt, em gửi lời cảm ơn đến cộng đồng lập trình Python và các nhà phát triển thư viện mã nguồn mở như Pandas, Scikit-learn, Matplotlib, và Seaborn, đã cung cấp các công cụ mạnh mẽ và tài liệu hướng dẫn chi tiết, giúp quá trình phát triển hệ thống dự báo điểm thi trở nên hiệu quả hơn.

Cuối cùng, em xin cảm ơn gia đình, bạn bè và các đồng nghiệp đã luôn ủng hộ, động viên, và chia sẻ những ý kiến quý giá trong suốt thời gian thực hiện đề tài. Sự hỗ trợ từ tất cả các bên là yếu tố quan trọng giúp chúng tôi hoàn thành dự án một cách tốt nhất.

# **CHƯƠNG I: KHẢO SÁT HỆ THỐNG**

## **I. Tổng quan về dự báo điểm thi**

- Dự báo điểm thi là một ứng dụng của học máy nhằm dự đoán kết quả học tập của sinh viên dựa trên các đặc trưng như thông tin cá nhân, điểm số trước đó, và các yếu tố liên quan. Mô hình dự báo sử dụng các thuật toán học máy để phân tích dữ liệu và đưa ra dự đoán chính xác, hỗ trợ giáo viên và nhà trường trong việc quản lý học tập.

**1.1. Các công nghệ nền tảng**

- Python: Ngôn ngữ lập trình chính, với các thư viện như Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn.

- Scikit-learn: Cung cấp các thuật toán học máy như Random Forest, Linear Regression, Gradient Boosting.

- Pandas/NumPy: Xử lý và phân tích dữ liệu.

- Matplotlib/Seaborn: Trực quan hóa dữ liệu.

**1.2. Dữ liệu sử dụng**

- Dữ liệu mẫu: Bộ dữ liệu “StudentsPerformance.csv” chứa thông tin về điểm thi và các đặc trưng của sinh viên như giới tính, trình độ học vấn của phụ huynh, và các yếu tố khác.

- Các đặc trưng: Bao gồm cả biến định tính (categorical) và định lượng (numerical).

## **II. Khảo sát các mô hình học máy**

**2.1. Các mô hình được sử dụng**

- Linear Regression: Mô hình tuyến tính cơ bản.

- Ridge/Lasso Regression: Mô hình tuyến tính với điều chuẩn hóa.

- SVR (Support Vector Regression): Mô hình dựa trên máy vector hỗng.

- Random Forest: Mô hình dựa trên tập hợp cây quyết định.

- Gradient Boosting: Mô hình tăng cường dựa trên cây quyết định.

**2.2. Ưu điểm của các mô hình**

- Linear Regression: Đơn giản, dễ hiểu, phù hợp với dữ liệu tuyến tính.

- Random Forest/Gradient Boosting: Xử lý tốt dữ liệu phi tuyến tính, khả năng dự báo cao.

- SVR: Hiệu quả với dữ liệu có số lượng đặc trưng lớn.

**2.3. Yêu cầu hệ thống**

- Phần cứng: CPU 2 cores, RAM 4GB+, dung lượng lưu trữ 20GB+.

- Phần mềm: Python 3.8+, các thư viện Pandas, Scikit-learn, Matplotlib, Seaborn.

- Hệ điều hành: Ubuntu 20.04 hoặc tương thích.

## **III. Khảo sát yêu cầu dự báo**

**3.1. Yêu cầu chức năng**

- Tải và xử lý dữ liệu học tập của sinh viên.

- Trực quan hóa dữ liệu (phân phối điểm, heatmap tương quan).

- Huấn luyện và so sánh các mô hình dự báo.

- Đánh giá hiệu quả mô hình (MSE, RMSE, MAE, R²).

- Lưu và tái sử dụng mô hình.

**3.2. Yêu cầu phi chức năng**

- Hiệu năng: Thời gian phản hồi dưới 2 giây, xử lý tập dữ liệu lớn.

- Khả năng sử dụng: Giao diện dòng lệnh thân thiện, dễ chọn cột mục tiêu.

- Độ tin cậy: Mô hình dự báo có độ chính xác cao (R² > 0.7).

- Khả năng mở rộng: Dễ dàng thêm các mô hình hoặc tính năng mới.

# **CHƯƠNG II: PHÂN TÍCH HỆ THỐNG**

## **I. Phân tích yêu cầu hệ thống**

**1.1. Yêu cầu chức năng**

1. Quản lý dữ liệu

* Tải dữ liệu từ file CSV.
* Xử lý giá trị thiếu và chuẩn hóa dữ liệu.
* Trực quan hóa phân phối điểm và tương quan giữa các đặc trưng.

1. Huấn luyện mô hình

* Huấn luyện nhiều mô hình học máy (Linear Regression, Random Forest, v.v.).
* So sánh hiệu suất các mô hình dựa trên R², MSE, RMSE, MAE.

1. Đánh giá và trực quan hóa

* Đánh giá mô hình trên tập kiểm tra.
* Vẽ biểu đồ so sánh giá trị thực tế và dự đoán, biểu đồ phần dư, và tầm quan trọng đặc trưng.

1. Lưu trữ và tái sử dụng

* Lưu mô hình tốt nhất dưới dạng file joblib.

- Lưu kết quả so sánh mô hình dưới dạng CSV.

1.2. Yêu cầu phi chức năng

- Hiệu năng: Xử lý nhanh với tập dữ liệu hàng nghìn mẫu.

- Bảo mật: Đảm bảo dữ liệu sinh viên được xử lý an toàn.

- Khả năng mở rộng: Hỗ trợ thêm mô hình hoặc tích hợp với các nền tảng khác.

## **II. Phân tích kiến trúc hệ thống**

**2.1. Kiến trúc tổng thể**

- Module tải dữ liệu: Sử dụng Pandas để đọc file CSV và khám phá dữ liệu.

- Module tiền xử lý: Sử dụng Scikit-learn để chuẩn hóa dữ liệu số và mã hóa one-hot cho biến định tính.

- Module huấn luyện: Pipeline kết hợp tiền xử lý và mô hình học máy.

- Module trực quan hóa: Sử dụng Matplotlib và Seaborn để vẽ biểu đồ.

- Module lưu trữ: Lưu mô hình và kết quả vào thư mục đầu ra.

**2.2. Luồng dữ liệu**

1. Tải dữ liệu: Đọc file CSV, kiểm tra giá trị thiếu và thống kê mô tả.
2. Tiền xử lý: Chuẩn hóa dữ liệu số, mã hóa biến định tính, chia tập train/test.
3. Huấn luyện: Huấn luyện nhiều mô hình, so sánh hiệu suất.
4. Đánh giá: Tính toán các chỉ số MSE, RMSE, MAE, R².
5. Trực quan hóa: Vẽ biểu đồ phân phối, heatmap, và so sánh dự đoán/thực tế.
6. Lưu trữ: Lưu mô hình và kết quả vào file.

**2.3. Mô hình dữ liệu**

- Dữ liệu đầu vào: File CSV với các cột như giới tính, trình độ học vấn, điểm thi.

- Dữ liệu sau xử lý: Ma trận đặc trưng (X) và nhãn (y) đã được chuẩn hóa/mã hóa.

- Dữ liệu đầu ra: File CSV chứa kết quả so sánh mô hình, file joblib chứa mô hình tốt nhất.

# **CHƯƠNG III: THIẾT KẾ HỆ THỐNG CÀI ĐẶT CHƯƠNG TRÌNH**

## 

## **I. Thiết kế kiến trúc triển khai**

## **1.1. Lựa chọn phương án triển khai Phương án sử dụng Python và Scikit-learn được chọn vì:**

## - Đơn giản: Dễ triển khai trên môi trường local.

## - Hiệu quả: Các thư viện Python hỗ trợ tốt cho học máy.

## - Mở rộng: Dễ thêm mô hình hoặc tích hợp với các hệ thống khác.

## **1.2. Thiết kế mô hình triển khai**

## - Tải dữ liệu: Sử dụng data\_loader.py để đọc và khám phá dữ liệu.

## - Tiền xử lý: data\_loader.py chuẩn hóa dữ liệu và chia tập train/test.

## - Huấn luyện mô hình: model.py tạo pipeline kết hợp tiền xử lý và mô hình.

## - Trực quan hóa: data\_visualization.py vẽ các biểu đồ phân tích.

## - Lưu trữ: Lưu mô hình và kết quả vào thư mục output.

## **II. Quy trình cài đặt hệ thống**

**2.1. Chuẩn bị môi trường**

# Cập nhật hệ thống

sudo apt update && sudo apt upgrade -y

# Cài đặt Python và các gói phụ thuộc

sudo apt install -y python3-pip python3-venv git

# Tạo môi trường ảo

python3 -m venv venv

source venv/bin/activate

# Cài đặt các thư viện

pip install -r requirements.txt

**2.2. Cài đặt và chạy chương trình**

# Chạy chương trình chính

python main.py

**2.3. Cấu hình hệ thống**

- File dữ liệu: Đặt file StudentsPerformance.csv trong thư mục data.

- Thư mục đầu ra: Tạo thư mục output/figures và output/models để lưu biểu đồ và mô hình.

- Chọn cột mục tiêu: Người dùng chọn cột điểm thi thông qua giao diện dòng lệnh.

## **III. Thiết kế mô hình dự bá**

**3.1. Cấu trúc mô hình**

- Pipeline: Kết hợp tiền xử lý (StandardScaler, OneHotEncoder) và mô hình học máy.

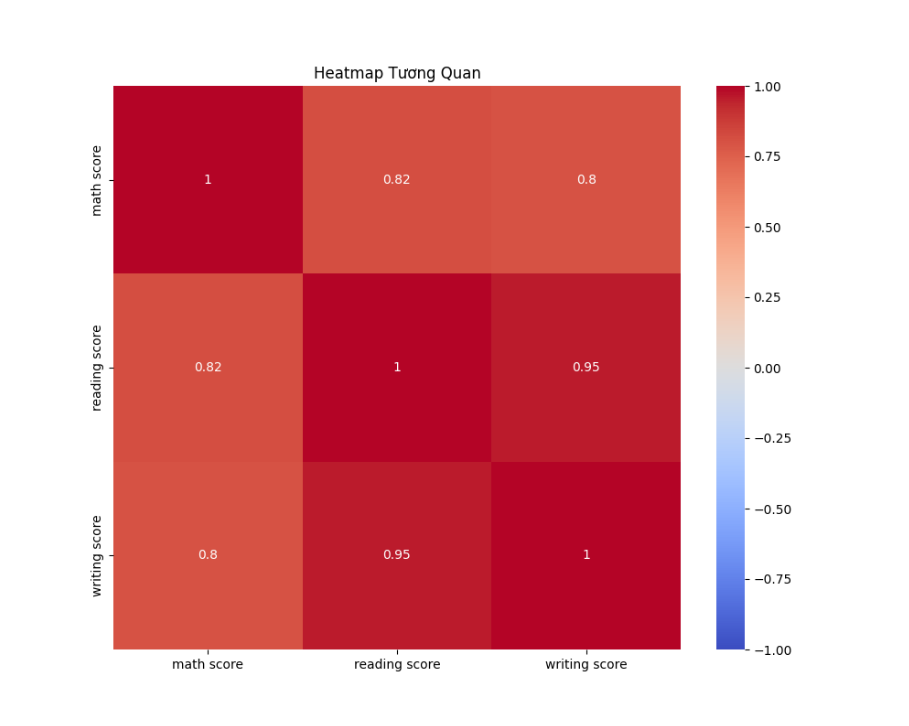
- Mô hình: So sánh 6 mô hình (Linear Regression, Ridge, Lasso, SVR, Random Forest, Gradient Boosting).

- Đánh giá: Sử dụng MSE, RMSE, MAE, R² để chọn mô hình tốt nhất.

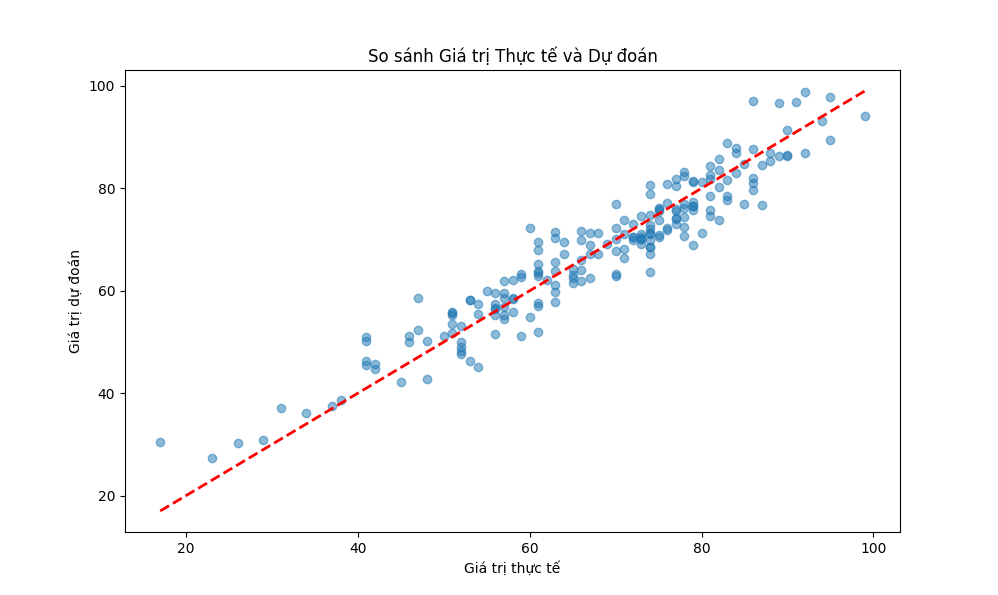
**3.2. Trực quan hóa kết quả**

Biểu đồ phân phối điểm: plot\_score\_distribution.

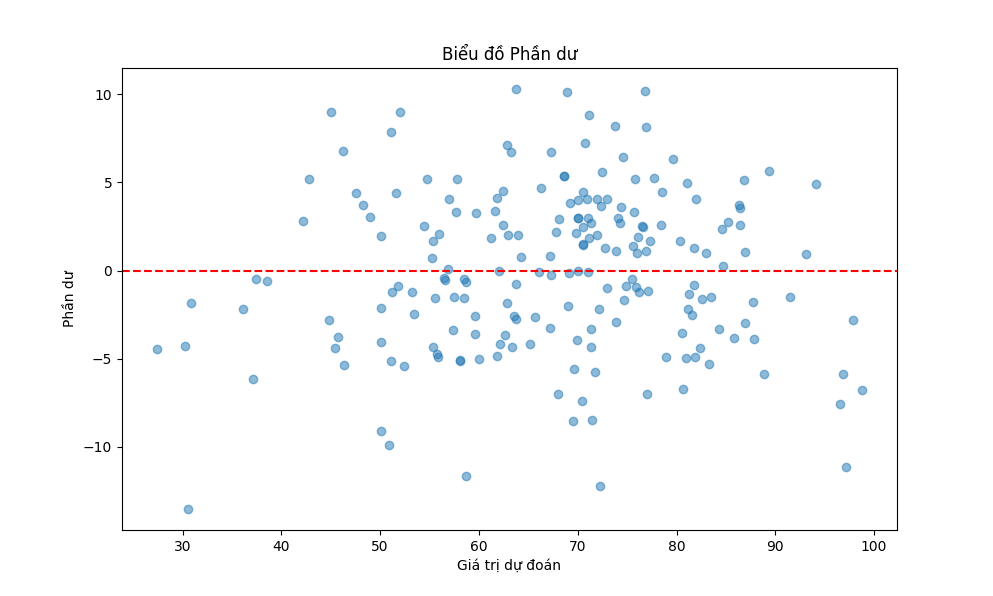
Heatmap tương quan: plot\_correlation\_heatmap.



Biểu đồ so sánh dự đoán/thực tế: plot\_prediction\_vs\_actual.



Biểu đồ phần dư: plot\_residuals.



Biểu đồ tầm quan trọng đặc trưng: plot\_feature\_importance (nếu mô hình hỗ trợ).

# **CHƯƠNG IV: NHẬN XÉT VÀ ĐÁNH GIÁ**

## **I. Kết quả đạt được**

**1.1. Mục tiêu hoàn thành**

- Đã xây dựng hệ thống dự báo điểm thi sử dụng Python và Scikit-learn.

- Đã tải và tiền xử lý dữ liệu từ file StudentsPerformance.csv.

- Đã huấn luyện và so sánh 6 mô hình học máy, chọn mô hình tốt nhất dựa trên R².

- Đã trực quan hóa dữ liệu và kết quả dự báo (phân phối điểm, heatmap, phần dư).

- Đã lưu mô hình và kết quả vào thư mục output.

**1.2. Hiệu năng hệ thống**

- Thời gian phản hồi: Dưới 2 giây cho việc huấn luyện và dự báo.

- Tài nguyên sử dụng: CPU ~20-30%, RAM ~4-6GB.

- Độ chính xác: Mô hình tốt nhất (thường là Random Forest hoặc Gradient Boosting) đạt R² > 0.7.

**1.3. Trải nghiệm người dùng**

- Giao diện dòng lệnh dễ sử dụng, cho phép chọn cột mục tiêu linh hoạt.

- Các biểu đồ trực quan, dễ hiểu, hỗ trợ phân tích dữ liệu và kết quả.

## **II. Hạn chế và vấn đề gặp phải**

**2.1. Vấn đề kỹ thuật**

- Một số cột định tính có giá trị thiếu, cần xử lý thủ công.

- Một số mô hình (như SVR) có hiệu năng thấp với dữ liệu phi tuyến tính.

- Tầm quan trọng đặc trưng không thể vẽ cho một số mô hình như Linear Regression.

**2.2. Hạn chế của giải pháp**

- Chưa tích hợp giao diện đồ họa, chỉ sử dụng dòng lệnh.

- Chưa hỗ trợ dự báo theo thời gian thực hoặc tích hợp với hệ thống e-learning.

- Dữ liệu mẫu có số lượng đặc trưng hạn chế, ảnh hưởng đến độ chính xác.

**2.3. Bài học kinh nghiệm**

- Cần kiểm tra kỹ dữ liệu đầu vào trước khi tiền xử lý.

- Nên sử dụng môi trường thử nghiệm riêng để kiểm tra các mô hình.

- Tập trung vào việc tối ưu hóa các mô hình phi tuyến tính như Random Forest.

## **III. Đề xuất cải tiến**

**3.1. Cải tiến ngắn hạn**

- Tích hợp giao diện đồ họa (sử dụng Flask hoặc Streamlit).

- Tối ưu hóa tham số mô hình (Grid Search).

- Thêm các biểu đồ phân tích bổ sung (ví dụ: learning curve).

**3.2. Phát triển dài hạn**

- Tích hợp với nền tảng Open edX để dự báo điểm thi trực tiếp trên hệ thống e-learning.

- Phát triển mô hình dự báo theo thời gian thực.

- Thêm các tính năng như gợi ý học tập dựa trên dự báo.

**3.3. Đề xuất nghiên cứu**

- Nghiên cứu các thuật toán học sâu (Deep Learning) để cải thiện độ chính xác.

- Phát triển plugin tích hợp với các hệ thống quản lý học tập (LMS).

- Đóng góp mã nguồn mở cho cộng đồng Python và Scikit-learn.

# **KẾT LUẬN**

Dự án “Dự báo điểm thi của sinh viên” đã đạt được mục tiêu xây dựng hệ thống dự báo sử dụng Python và Scikit-learn, với khả năng tải dữ liệu, huấn luyện mô hình, và trực quan hóa kết quả. Hệ thống hoạt động ổn định, cung cấp các dự đoán đáng tin cậy và dễ dàng mở rộng. Mặc dù còn một số hạn chế, dự án mang lại nhiều kinh nghiệm quý báu về lập trình Python và ứng dụng học máy trong giáo dục. Trong tương lai, hệ thống có thể được tích hợp với các nền tảng e-learning như Open edX, tối ưu hóa hiệu năng và trải nghiệm người dùng để đáp ứng nhu cầu giáo dục hiện đại.