**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT**



**BÀI TẬP LỚN**

**MÔN: KỸ NGHỆ TRI THỨC VÀ HỌC MÁY**

**TÊN ĐỀ TÀI: DỰ ĐOÁN ĐIỂM HỌC SINH**

**Thành viên nhóm:**

* + - * 1. **PHAN THANH TÚ (2221050845)**
        2. **NGUYỄN ĐÌNH TRÁNG (2221050306)**
        3. **TRẦN ĐĂNG QUANG (2221050138)**

**Hà Nội, Năm 2025**

# **LỜI CẢM ƠN**

Trong quá trình thực hiện đồ án tốt nghiệp, chúng em đã nhận được sự giúp đỡ, đóng góp ý kiến và sự hướng dẫn tận tình từ các thầy cô và bạn bè.

Chúng em xin gửi lời cảm ơn chân thành đến thầy **Nguyễn Văn Thắng** và cô **Nguyễn Thị Mai Dung** – giảng viên Bộ môn Kỹ nghệ Tri thức và Học máy, Trường Đại học Mỏ Địa Chất – những người đã tận tình hướng dẫn và chỉ bảo chúng em trong suốt quá trình thực hiện Bài Tập Lớn.

Chúng em cũng xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong Trường Đại học Mỏ Địa Chất, đặc biệt là các thầy cô trong chuyên ngành Công Nghệ Thông Tin chất lượng cao, những người đã truyền đạt cho chúng em những kiến thức quý báu về đại cương và chuyên ngành, giúp chúng em có nền tảng lý thuyết vững vàng, đồng thời tạo điều kiện thuận lợi trong suốt quá trình học tập và thực hiện bài tập lớn.

Cuối cùng, chúng em xin chân thành cảm ơn bạn bè, thầy cô – những người luôn ở bên cạnh, quan tâm, động viên và giúp đỡ chúng em trong suốt quá trình học tập và hoàn thành đồ án.

Do thời gian thực hiện có hạn và phạm vi nghiên cứu rộng, cùng với kinh nghiệm còn hạn chế, đồ án này không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý, chỉ bảo từ thầy cô để có thể bổ sung và hoàn thiện kiến thức của mình, nhằm phục vụ tốt hơn cho công tác thực tế sau này.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[1. Giới thiệu đề tài 1](#_Toc201070286)

[1.1 Lý do nghiên cứu 1](#_Toc201070287)

[1.2 Mục tiêu của dự án 1](#_Toc201070288)

[1.3 Tầm quan trọng của dự án 1](#_Toc201070289)

[1.4 Phạm vi nghiên cứu 2](#_Toc201070290)

[2. Tổng quan lý thuyết 2](#_Toc201070291)

[2.1 Hồi quy tuyến tính đa biến (Multiple Linear Regression) 2](#_Toc201070292)

[2.2 Gradient Descent 3](#_Toc201070293)

[2.3 Lí do chọn thuật toán 4](#_Toc201070294)

[3. Thiết kế hệ thống 4](#_Toc201070295)

[3.1 Kiến trúc hệ thống tổng quan 4](#_Toc201070296)

[3.2 Sơ đồ triển khai hệ thống (Deployment Diagram) 5](#_Toc201070297)

[3.3 Cấu trúc dữ liệu 5](#_Toc201070298)

[3.4 Mô hình dữ liệu 5](#_Toc201070299)

[4. Quá trình huấn luyện mô hình 6](#_Toc201070300)

[4.1 Tiền xử lý dữ liệu 11](#_Toc201070301)

[4.2 Phân chia dữ liệu 12](#_Toc201070303)

[4.3 Huấn luyện mô hình hồi quy tuyến tính 13](#_Toc201070306)

[4.4 Đánh giá mô hình 15](#_Toc201070308)

[5. Triển khai hệ thống 16](#_Toc201070311)

[5.1 Backend với Flask 16](#_Toc201070312)

[5.2 Frontend với React 18](#_Toc201070315)

[5.3 Kiểm thử hệ thống 20](#_Toc201070317)

[6. Đánh giá kết quả 22](#_Toc201070320)

[6.1 Kêt quả thực nghiệm 22](#_Toc201070321)

[6.2 Đánh giá trải nghiệm người dùng 23](#_Toc201070322)

[7. Định hướng phát triển trong tương lai 23](#_Toc201070323)

[7.1 Cải thiện mô hình 23](#_Toc201070324)

[7.2 Mở rộng ứng dụng 24](#_Toc201070325)

[7.3 Tối ưu hóa hiệu suất 24](#_Toc201070326)

[8. Kết luận 25](#_Toc201070327)

[9. Tài liệu tham khảo 25](#_Toc201070328)

[10. Phụ lục 26](#_Toc201070329)

[10.1 Code nguồn 26](#_Toc201070330)

[10.2 Giao diện ứng dụng 26](#_Toc201070331)

## **1. Giới thiệu đề tài**

### **1.1 Lý do nghiên cứu**

Trong bối cảnh giáo dục ngày nay, việc dự đoán kết quả học tập của học sinh là một vấn đề quan trọng đối với không chỉ các giáo viên mà còn đối với phụ huynh và các nhà quản lý giáo dục. Một trong những vấn đề mà nhiều phụ huynh gặp phải là không thể biết trước kết quả học tập của con em mình, điều này khiến cho việc định hướng học tập và hỗ trợ học sinh gặp khó khăn. Vì vậy, việc có thể dự đoán điểm số của học sinh dựa trên các yếu tố đầu vào đơn giản sẽ giúp phụ huynh và giáo viên có được những thông tin quan trọng để đưa ra những quyết định phù hợp.

Dự án của chúng tôi nhằm phát triển một hệ thống có thể dự đoán điểm số học sinh dựa trên một số thuộc tính cơ bản như thời gian học, tình trạng gia đình, sức khỏe, và các điểm số trước đó. Hệ thống này không chỉ giúp phụ huynh có cái nhìn rõ ràng hơn về khả năng học tập của con em mình mà còn giúp các giáo viên đưa ra những biện pháp hỗ trợ kịp thời.

### **1.2 Mục tiêu của dự án**

Mục tiêu chính của dự án là xây dựng một mô hình dự đoán điểm số học sinh, sử dụng thuật toán hồi quy tuyến tính đa biến. Với các dữ liệu đầu vào như thời gian học (studytime), số lần thất bại (failures), mức độ quan hệ gia đình (famrel), sức khỏe (health), việc có Internet (internet), số ngày vắng mặt (absences), tình trạng gia đình hỗ trợ (famsup), sự hỗ trợ của trường học (schoolsup), cùng với điểm số các kỳ học trước (G1 và G2), hệ thống sẽ có thể tính toán và dự đoán điểm số cuối kỳ (G3) cho học sinh.

Mục tiêu cụ thể của dự án là:

* Phát triển một mô hình học máy có thể dự đoán điểm số học sinh một cách nhanh chóng và chính xác.
* Cung cấp một công cụ đơn giản và dễ sử dụng cho phụ huynh và giáo viên để theo dõi và hỗ trợ học sinh.

### **1.3 Tầm quan trọng của dự án**

Dự án này có tầm quan trọng lớn đối với giáo dục, đặc biệt là trong việc cải thiện cách thức hỗ trợ học sinh. Việc dự đoán điểm số học sinh có thể giúp phụ huynh đưa ra những quyết định tốt hơn trong việc hỗ trợ con em mình học tập. Các giáo viên cũng có thể sử dụng công cụ này để xác định những học sinh có thể cần sự hỗ trợ đặc biệt, từ đó đưa ra các biện pháp can thiệp kịp thời.

Ngoài ra, dự án này cũng đóng góp vào việc áp dụng các công nghệ học máy trong giáo dục, tạo ra các giải pháp tiên tiến giúp nâng cao chất lượng giảng dạy và học tập. Hệ thống dự đoán điểm số cũng có thể trở thành một công cụ hữu ích trong việc tối ưu hóa các chiến lược giảng dạy và phát triển chương trình học.

### **1.4 Phạm vi nghiên cứu**

Phạm vi nghiên cứu của dự án này tập trung vào việc sử dụng các dữ liệu có sẵn từ bộ dữ liệu điểm số học sinh, được thu thập từ các nguồn tài liệu trực tuyến và các nghiên cứu trước đó. Dữ liệu đầu vào sẽ bao gồm các yếu tố đơn giản như thời gian học, tình trạng gia đình, và các điểm số học kỳ trước của học sinh, và mục tiêu là dự đoán điểm số cuối kỳ (G3).

Phạm vi nghiên cứu của dự án chủ yếu bao gồm:

* Dữ liệu có sẵn: Bộ dữ liệu học sinh sẽ được sử dụng để huấn luyện mô hình. Dữ liệu này đã được thu thập và có sẵn từ các nghiên cứu trước đây.
* Kiến thức học máy cơ bản: Dự án này sẽ áp dụng các kiến thức học máy cơ bản, đặc biệt là về thuật toán hồi quy tuyến tính, để xây dựng mô hình dự đoán.
* Tự học và nghiên cứu: Phần lớn kiến thức về học máy và kỹ thuật mô hình hóa sẽ được tự học thông qua các tài liệu nghiên cứu và khóa học trực tuyến.

## **2. Tổng quan lý thuyết**

### **2.1 Hồi quy tuyến tính đa biến (Multiple Linear Regression)**

Hồi quy tuyến tính đa biến (Multiple Linear Regression - MLR) là một mô hình học máy được sử dụng để dự đoán một biến phụ thuộc (dependent variable) từ một hoặc nhiều biến độc lập (independent variables). Đây là một mở rộng của hồi quy tuyến tính đơn biến (Simple Linear Regression), nơi mối quan hệ giữa biến phụ thuộc và các biến độc lập được mô tả bằng một hàm tuyến tính.

Công thức tổng quát của mô hình hồi quy tuyến tính đa biến là:

A black screen with white text

AI-generated content may be incorrect.

Các giả định cơ bản của hồi quy tuyến tính đa biến:

1. Mối quan hệ tuyến tính: Giả định rằng mối quan hệ giữa biến phụ thuộc và các biến độc lập là tuyến tính.
2. Độc lập giữa các sai số: Các sai số (residuals) giữa các quan sát phải độc lập với nhau.
3. Phân phối chuẩn của sai số: Sai số của mô hình nên tuân theo phân phối chuẩn.
4. Đồng nhất phương sai (Homoscedasticity): Phương sai của sai số phải là hằng số, tức là không phụ thuộc vào giá trị của các biến độc lập.

**Ví dụ minh họa**:

Giả sử chúng ta muốn dự đoán điểm số cuối kỳ G3của học sinh dựa trên các yếu tố như thời gian học (studytime), số lần thất bại (failuresfailuresfailures), và điểm các kỳ học trước (G1, G2). Mô hình hồi quy tuyến tính đa biến có thể được biểu diễn như sau:



Chúng ta sẽ huấn luyện mô hình để tìm ra các trọng số w1, w2, w3, w4 và bias b sao cho độ chính xác của mô hình là cao nhất.

### **2.2 Gradient Descent**

**Gradient Descent** là một thuật toán tối ưu hóa được sử dụng để tìm giá trị tối ưu của các tham số (trọng số và hệ số chệch) trong các mô hình học máy, bao gồm hồi quy tuyến tính. Mục tiêu của thuật toán là tối thiểu hóa hàm chi phí (cost function) bằng cách điều chỉnh các tham số theo hướng giảm dần của gradient (đạo hàm).

**Khái niệm lý thuyết**:

Thuật toán Gradient Descent cập nhật các trọng số www và hệ số chệch bbb qua từng bước lặp, sao cho hàm chi phí (cost function) đạt giá trị nhỏ nhất. Hàm chi phí trong trường hợp của hồi quy tuyến tính là **Mean Squared Error (MSE)**, được tính như sau:

A black background with white text

AI-generated content may be incorrect.

**Cập nhật trọng số**:

Để tối thiểu hóa hàm chi phí, Gradient Descent sẽ tính toán gradient của hàm chi phí theo các trọng số và hệ số chệch, sau đó cập nhật các tham số theo công thức sau:

A black background with white text

AI-generated content may be incorrect.

Thuật toán sẽ tiếp tục cập nhật các trọng số và hệ số chệch qua các bước lặp cho đến khi hàm chi phí hội tụ (không thay đổi nhiều giữa các lần lặp).

### **2.3 Lí do chọn thuật toán**

Hồi quy tuyến tính đa biến (Multiple Linear Regression) là lựa chọn tối ưu cho bài toán dự đoán điểm số học sinh vì một số lý do sau:

1. **Mô hình đơn giản và dễ hiểu**: Hồi quy tuyến tính đa biến cung cấp một mô hình dễ hiểu với các trọng số rõ ràng, giúp phụ huynh và giáo viên dễ dàng theo dõi các yếu tố ảnh hưởng đến điểm số của học sinh.
2. **Dữ liệu đầu vào đơn giản**: Dữ liệu trong bài toán của bạn chủ yếu bao gồm các yếu tố như thời gian học (studytime), số lần thất bại (failure), mức độ quan hệ gia đình (famrel), sức khỏe (health), việc có Internet (internet), số ngày vắng mặt (absences), tình trạng gia đình hỗ trợ (famsup), sự hỗ trợ của trường học (schoolsup), và điểm số trước đó (G1, G2), các yếu tố này có thể được mô tả tốt bằng mô hình hồi quy tuyến tính.
3. **Hiệu suất tốt với dữ liệu tuyến tính**: Hồi quy tuyến tính đa biến rất hiệu quả khi có mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc. Trong trường hợp này, các yếu tố như thời gian học, điểm số trước đó có thể có ảnh hưởng tuyến tính đến điểm số cuối kỳ của học sinh.
4. **Tính khả thi**: Hồi quy tuyến tính đa biến yêu cầu ít tài nguyên tính toán và dễ dàng triển khai, giúp tiết kiệm thời gian và tài nguyên cho việc huấn luyện và triển khai hệ thống.

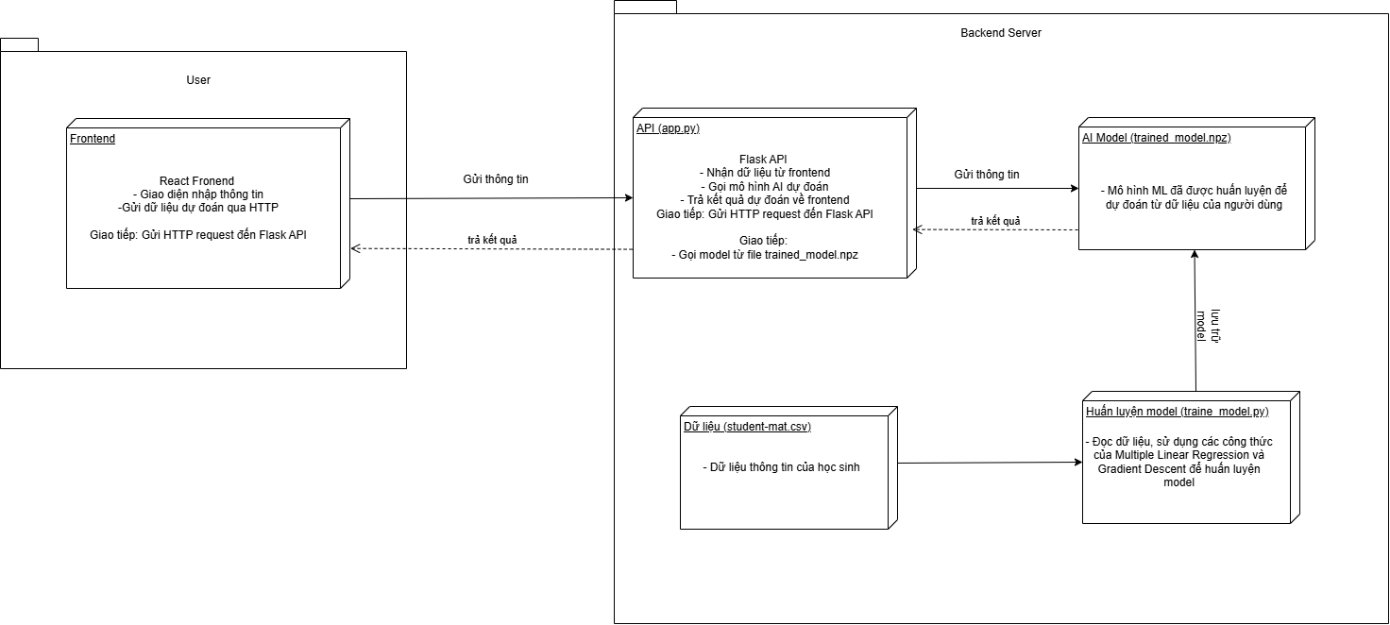
## **3. Thiết kế hệ thống**

### **3.1 Kiến trúc hệ thống tổng quan**

Hệ thống được chia thành ba thành phần chính:

* **Frontend (ReactJS):** Giao diện người dùng cho phép nhập dữ liệu đầu vào và hiển thị kết quả dự đoán.
* **Backend (Flask):** Xử lý yêu cầu từ frontend, thực hiện dự đoán bằng mô hình học máy.
* **Mô hình học máy:** Được huấn luyện từ trước và lưu trữ dưới dạng file để sử dụng khi có yêu cầu dự đoán.

### **3.2 Sơ đồ triển khai hệ thống (Deployment Diagram)**



### **3.3 Cấu trúc dữ liệu**

Dữ liệu đầu vào có định dạng JSON với các trường như:

{  
 "studytime": 2,  
 "failures": 1,  
 "famrel": 4,  
 "health": 5,  
 "internet": 1,  
 "absences": 3,  
 "famsup": 1,  
 "schoolsup": 0,  
 "G1": 12,  
 "G2": 13  
}

### **3.4 Mô hình dữ liệu**

Mô hình học máy sử dụng hồi quy tuyến tính đa biến (Multiple Linear Regression) với các biến độc lập như trên để dự đoán biến phụ thuộc là G3 – điểm số cuối kỳ.

## **4. Quá trình huấn luyện mô hình**

Dữ liệu ban đầu gồm có các thuộc tính sau

1. **school** - Trường học của học sinh (dạng nhị phân):
   1. 'GP' – Gabriel Pereira
   2. 'MS' – Mousinho da Silveira
2. **sex** - Giới tính của học sinh (dạng nhị phân):
   1. 'F' – Nữ
   2. 'M' – Nam
3. **age** - Tuổi của học sinh (dạng số):
   1. Từ 15 đến 22 tuổi
4. **address** - Loại địa chỉ nhà của học sinh (dạng nhị phân):
   1. 'U' – Thành thị
   2. 'R' – Nông thôn
5. **famsize** - Quy mô gia đình (dạng nhị phân):
   1. 'LE3' – Gia đình có ít hơn hoặc bằng 3 thành viên
   2. 'GT3' – Gia đình có hơn 3 thành viên
6. **Pstatus** - Tình trạng sống chung của cha mẹ (dạng nhị phân):
   1. 'T' – Cùng sống
   2. 'A' – Xa nhau
7. **Medu** - Trình độ học vấn của mẹ (dạng số):
   1. 0 – Không có học vấn
   2. 1 – Tiểu học (lớp 4)
   3. 2 – Lớp 5 đến 9
   4. 3 – Trung học
   5. 4 – Giáo dục đại học
8. **Fedu** - Trình độ học vấn của cha (dạng số):
   1. 0 – Không có học vấn
   2. 1 – Tiểu học (lớp 4)
   3. 2 – Lớp 5 đến 9
   4. 3 – Trung học
   5. 4 – Giáo dục đại học
9. **Mjob** - Nghề nghiệp của mẹ (dạng danh mục):
   1. 'teacher' – Giáo viên
   2. 'health' – Ngành y tế
   3. 'services' – Dịch vụ (như hành chính hoặc công an)
   4. 'at\_home' – Nội trợ
   5. 'other' – Nghề nghiệp khác
10. **Fjob** - Nghề nghiệp của cha (dạng danh mục):
    1. 'teacher' – Giáo viên
    2. 'health' – Ngành y tế
    3. 'services' – Dịch vụ (như hành chính hoặc công an)
    4. 'at\_home' – Nội trợ
    5. 'other' – Nghề nghiệp khác
11. **reason** - Lý do chọn trường (dạng danh mục):
    1. 'home' – Gần nhà
    2. 'reputation' – Danh tiếng trường
    3. 'course' – Sở thích môn học
    4. 'other' – Lý do khác
12. **guardian** - Người giám hộ của học sinh (dạng danh mục):
    1. 'mother' – Mẹ
    2. 'father' – Cha
    3. 'other' – Người khác
13. **traveltime** - Thời gian di chuyển từ nhà đến trường (dạng số):
    1. 1 – <15 phút
    2. 2 – 15 đến 30 phút
    3. 3 – 30 phút đến 1 giờ
    4. 4 – >1 giờ
14. **studytime** - Thời gian học tập mỗi tuần (dạng số):
    1. 1 – <2 giờ
    2. 2 – 2 đến 5 giờ
    3. 3 – 5 đến 10 giờ
    4. 4 – >10 giờ
15. **failures** - Số lần học sinh đã trượt môn (dạng số):
    1. 1 <= n < 3, nếu không thì là 4
16. **schoolsup** - Hỗ trợ giáo dục thêm (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
17. **famsup** - Hỗ trợ giáo dục từ gia đình (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
18. **paid** - Lớp học phụ đạo trả phí (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
19. **activities** - Các hoạt động ngoại khóa (dạng nhị phân):
    1. 'yes' – Có tham gia
    2. 'no' – Không tham gia
20. **nursery** - Tham gia trường mẫu giáo (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
21. **higher** - Muốn học đại học (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
22. **internet** - Có Internet tại nhà (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
23. **romantic** - Đang có mối quan hệ tình cảm (dạng nhị phân):
    1. 'yes' – Có
    2. 'no' – Không
24. **famrel** - Chất lượng mối quan hệ gia đình (dạng số):
    1. Từ 1 (rất tệ) đến 5 (rất tốt)
25. **freetime** - Thời gian rảnh sau giờ học (dạng số):
    1. Từ 1 (rất ít) đến 5 (rất nhiều)
26. **goout** - Thời gian ra ngoài với bạn bè (dạng số):
    1. Từ 1 (rất ít) đến 5 (rất nhiều)
27. **Dalc** - Tiêu thụ rượu trong các ngày làm việc (dạng số):
    1. Từ 1 (rất ít) đến 5 (rất nhiều)
28. **Walc** - Tiêu thụ rượu vào cuối tuần (dạng số):
    1. Từ 1 (rất ít) đến 5 (rất nhiều)
29. **health** - Tình trạng sức khỏe hiện tại (dạng số):
    1. Từ 1 (rất tệ) đến 5 (rất tốt)
30. **absences** - Số lần học sinh vắng mặt (dạng số):
    1. Từ 0 đến 93

Các điểm số này liên quan đến môn học mà học sinh đang theo học, bao gồm:

* **G1** - Điểm kỳ 1 (dạng số): Từ 0 đến 20
* **G2** - Điểm kỳ 2 (dạng số): Từ 0 đến 20
* **G3** - Điểm cuối kỳ (dạng số): Từ 0 đến 20 (đây là mục tiêu dự đoán)

Nhưng đối với ứng dụng chỉ lựa chọn các thuộc tính như studytime, failures, famrel, health, internet, absences, famsup, schoolsup, G1, G2 để dự đoán điểm G3 với các lí dó sau:

**studytime** (Thời gian học tập hàng tuần):

* Thời gian học tập của học sinh có ảnh hưởng trực tiếp đến kết quả học tập của họ. Học sinh dành nhiều thời gian cho việc học sẽ có xu hướng đạt điểm cao hơn. Do đó, đây là một yếu tố quan trọng trong việc dự đoán điểm số cuối kỳ (G3).

**failures** (Số lần học sinh trượt môn):

* Số lần trượt môn trong quá khứ phản ánh khả năng học tập của học sinh. Học sinh có nhiều lần trượt môn có thể gặp khó khăn trong việc cải thiện kết quả học tập trong các kỳ sau. Thuộc tính này giúp nhận diện các học sinh có nguy cơ thấp trong việc đạt điểm cao.

**famrel** (Chất lượng mối quan hệ gia đình):

* Mối quan hệ gia đình ảnh hưởng đến trạng thái tinh thần và sự ổn định của học sinh. Một gia đình có mối quan hệ tốt sẽ tạo ra môi trường hỗ trợ và khuyến khích học sinh học tập tốt hơn, từ đó ảnh hưởng tích cực đến điểm số của họ.

**health** (Tình trạng sức khỏe):

* Sức khỏe là yếu tố ảnh hưởng đến khả năng tập trung và sự tham gia vào học tập của học sinh. Học sinh có sức khỏe tốt có thể học tập hiệu quả hơn và ít bị gián đoạn trong quá trình học, từ đó cải thiện điểm số cuối kỳ.

**internet** (Truy cập Internet tại nhà):

* Việc có internet tại nhà sẽ tạo điều kiện cho học sinh truy cập tài liệu học tập trực tuyến và tham gia vào các khóa học bổ sung. Điều này giúp nâng cao khả năng học tập và khả năng tiếp cận các nguồn tài liệu cần thiết, từ đó có thể cải thiện điểm số.

**absences** (Số lần vắng mặt):

* Số lần vắng mặt của học sinh có thể là chỉ số quan trọng về mức độ tham gia và cam kết học tập của học sinh. Những học sinh vắng mặt nhiều có thể bỏ lỡ các bài học quan trọng, điều này có thể ảnh hưởng tiêu cực đến kết quả học tập của họ.

**famsup** (Hỗ trợ giáo dục từ gia đình):

* Hỗ trợ giáo dục từ gia đình đóng vai trò quan trọng trong việc khuyến khích học sinh học tập. Khi gia đình tham gia tích cực vào quá trình học của học sinh, học sinh có thể có sự động viên và hỗ trợ cần thiết để nâng cao kết quả học tập.

**schoolsup** (Hỗ trợ giáo dục thêm từ trường):

* Sự hỗ trợ giáo dục thêm từ trường (như các lớp học thêm, các buổi học phụ đạo) có thể giúp học sinh nâng cao kiến thức và cải thiện điểm số trong kỳ thi cuối kỳ. Học sinh nhận được sự hỗ trợ này thường có cơ hội đạt điểm cao hơn.

**G1 và G2** (Điểm kỳ 1 và kỳ 2):

* Điểm số trong kỳ 1 và kỳ 2 là yếu tố quan trọng để dự đoán điểm số cuối kỳ (G3). Những học sinh có điểm số cao trong các kỳ trước có thể duy trì được kết quả học tập tốt, và điều này giúp dự đoán điểm số cuối kỳ chính xác hơn.

### **4.1 Tiền xử lý dữ liệu**

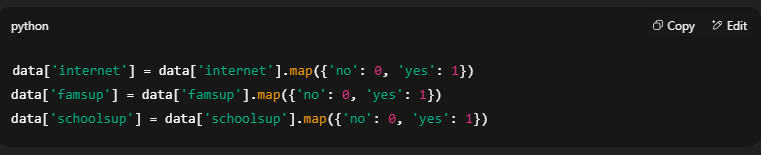
Tiền xử lý dữ liệu là bước quan trọng giúp chuẩn bị đầu vào phù hợp cho mô hình học máy. Trong dự án này, dữ liệu được lấy từ tập tin student-mat.csv và xử lý trong hàm preprocess\_data() của file train\_model.py. Quá trình tiền xử lý bao gồm các bước cụ thể sau:

### **a) Mã hóa dữ liệu nhị phân**

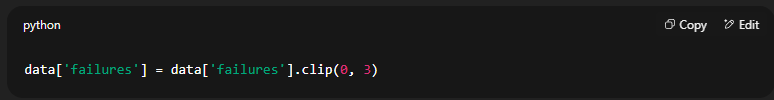
Một số cột dữ liệu có giá trị dạng nhị phân "yes" hoặc "no", điển hình như:

* internet: học sinh có truy cập Internet tại nhà hay không,
* famsup: có được hỗ trợ học tập từ gia đình,
* schoolsup: có được hỗ trợ học tập từ trường.

Những cột này được mã hóa sang số (1 hoặc 0) để mô hình có thể hiểu được. Cụ thể:

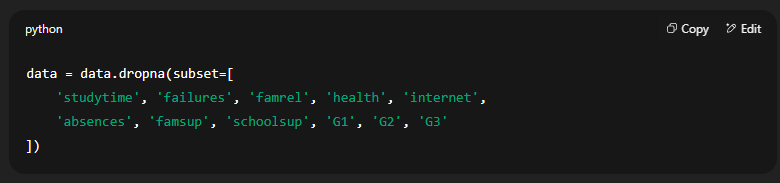
**b) Giới hạn dữ liệu bất thường**

Biến failures (số lần trượt môn) được giới hạn trong khoảng từ 0 đến 3 nhằm tránh các giá trị ngoại lai không hợp lý:

**c) Xử lý giá trị thiếu**

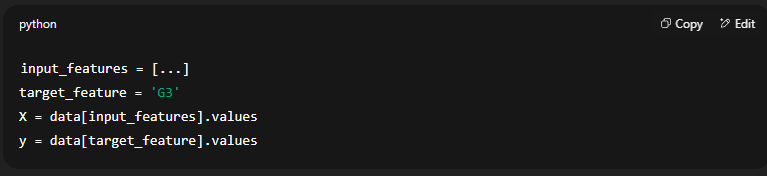
Mô hình yêu cầu đầy đủ các giá trị cho các cột quan trọng. Do đó, dữ liệu sẽ loại bỏ các dòng có chứa giá trị thiếu (NaN) tại các cột sau:

* studytime, failures, famrel, health, internet, absences,
* famsup, schoolsup, G1, G2, G3.

**d) Tách đặc trưng và mục tiêu dự đoán**

Sau khi dữ liệu đã được xử lý, các biến đầu vào (features) và biến đầu ra (target) được tách riêng để đưa vào mô hình học máy:

* **Các đặc trưng đầu vào gồm:** ['studytime', 'failures', 'famrel', 'health', 'internet', 'absences', 'famsup', 'schoolsup', 'G1', 'G2']
* **Biến mục tiêu:** G3 – điểm tổng kết cuối kỳ của học sinh.



### **4.2 Phân chia dữ liệu**

Sau khi hoàn tất quá trình tiền xử lý, dữ liệu cần được chia thành hai phần riêng biệt để huấn luyện và đánh giá mô hình. Việc phân chia này nhằm đảm bảo rằng mô hình được kiểm tra trên dữ liệu “chưa từng thấy”, giúp đánh giá đúng khả năng tổng quát hóa của mô hình.

### **a) Mục đích phân chia**

Việc tách dữ liệu thành hai tập:

* **Tập huấn luyện (training set):** dùng để mô hình học các mối quan hệ giữa đầu vào và đầu ra.
* **Tập kiểm tra (test set):** dùng để đánh giá hiệu quả của mô hình sau khi huấn luyện.

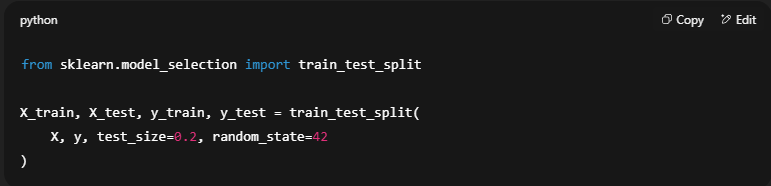
Nếu không có bước này, mô hình có thể **học thuộc lòng dữ liệu** mà không thật sự hiểu bản chất, dẫn đến hiện tượng **overfitting** – dự đoán tốt trên tập huấn luyện nhưng kém trên dữ liệu thực tế.

### **b) Tỷ lệ phân chia**

Trong dự án này, dữ liệu được chia theo tỷ lệ:

* **80% cho tập huấn luyện**
* **20% cho tập kiểm tra**

Việc phân chia được thực hiện bằng hàm train\_test\_split() từ thư viện scikit-learn:

Giải thích:

* X và y là các đặc trưng và nhãn sau khi tiền xử lý.
* test\_size=0.2 chỉ định 20% dữ liệu được dành cho kiểm tra.
* random\_state=42 đảm bảo việc chia tách có thể lặp lại (tái hiện được kết quả).

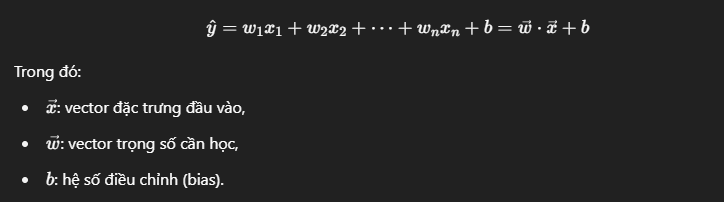
Sau bước này, mô hình sẵn sàng để được huấn luyện bằng tập huấn luyện X\_train, y\_train.

### **4.3 Huấn luyện mô hình hồi quy tuyến tính**

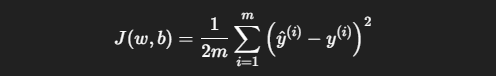
Sau khi đã có tập huấn luyện (X\_train, y\_train), bước tiếp theo là xây dựng và huấn luyện mô hình để dự đoán điểm tổng kết cuối kỳ (G3) của học sinh. Mô hình được sử dụng là **hồi quy tuyến tính bội (Multiple Linear Regression)** – một thuật toán cơ bản nhưng hiệu quả trong bài toán dự đoán giá trị số.

### **a) Mô hình hồi quy tuyến tính bội**

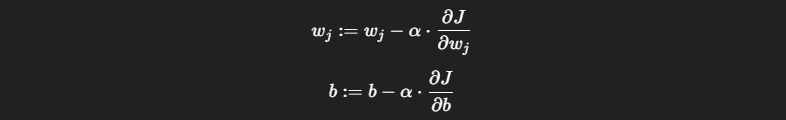
Hồi quy tuyến tính bội dự đoán giá trị đầu ra *y^\hat{y}*y^ bằng công thức:

**b) Tối ưu trọng số bằng Gradient Descent**

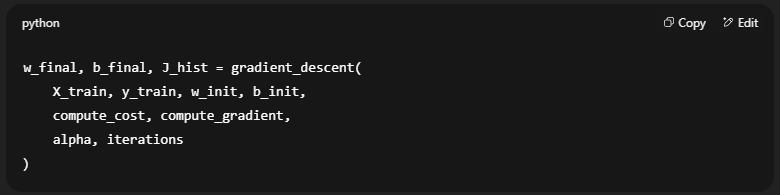
Mục tiêu là tìm các trọng số *w⃗\vec{w}*w và *bb*b sao cho sai số giữa dự đoán và thực tế nhỏ nhất. Sai số được đo bằng **hàm chi phí bình phương trung bình (Mean Squared Error – MSE)**:



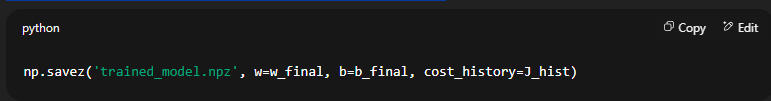
Thuật toán **Gradient Descent** được sử dụng để tối thiểu hóa hàm chi phí. Ở mỗi vòng lặp, mô hình cập nhật trọng số theo đạo hàm:

Trong mã nguồn (utils.py), điều này được hiện thực hóa bằng hàm gradient\_descent() với các tham số:

* alpha = 0.003: tốc độ học,
* iterations = 10000: số vòng lặp cập nhật.

**c) Lưu mô hình sau huấn luyện**

Sau khi hoàn tất huấn luyện, mô hình với các trọng số tối ưu *w⃗,b\vec{w}, b*w,b sẽ được lưu lại bằng thư viện numpy để sử dụng lại khi triển khai (không cần huấn luyện lại mỗi lần chạy):

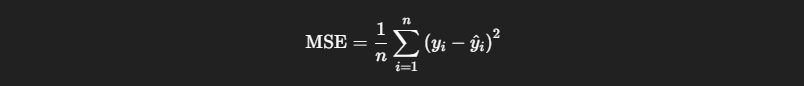
File trained\_model.npz chứa:

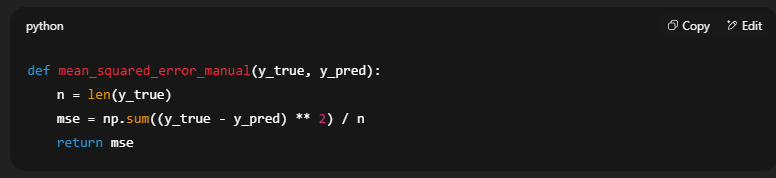
* Trọng số đã học w\_final,
* Hệ số điều chỉnh b\_final,
* Lịch sử hàm chi phí (cost\_history) – để theo dõi độ hội tụ.

### **4.4 Đánh giá mô hình**

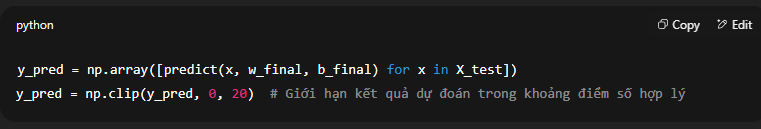
* Sau khi huấn luyện mô hình trên tập dữ liệu huấn luyện, bước tiếp theo là **đánh giá mức độ chính xác và khả năng tổng quát hóa** của mô hình trên tập dữ liệu kiểm tra (X\_test, y\_test).
* Việc đánh giá này giúp xác định xem mô hình có thật sự học được bản chất của dữ liệu hay chỉ đơn giản là ghi nhớ dữ liệu huấn luyện (overfitting).

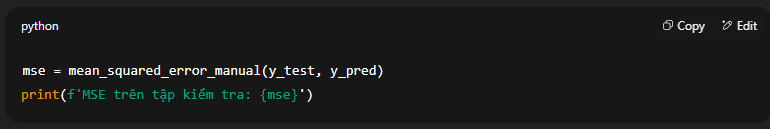
### **a) Chỉ số đánh giá: Mean Squared Error (MSE)**

* MSE (Mean Squared Error – sai số bình phương trung bình) là chỉ số phổ biến dùng để đo lường sai lệch giữa giá trị dự đoán và giá trị thực tế:
* 
* Trong mã nguồn, MSE được tính thủ công bằng hàm:

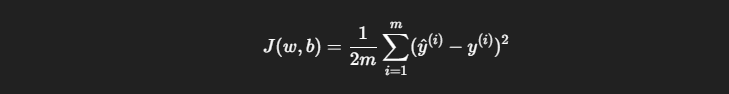
**b) Dự đoán trên tập kiểm tra**

Mô hình sau huấn luyện được dùng để dự đoán điểm số G3 trên tập kiểm tra:

Sau đó, MSE được tính bằng:

**c) Hàm chi phí (Cost Function)**

Ngoài MSE, mô hình cũng theo dõi **giá trị hàm chi phí (J)** trong suốt quá trình huấn luyện bằng công thức:



hác biệt duy nhất giữa MSE và J(w, b) là hệ số chia 2. Giá trị J được lưu vào danh sách cost\_history để theo dõi độ hội tụ của mô hình theo thời gian.

### **Kết luận**

Thông qua chỉ số **MSE** và **giá trị hàm chi phí**, nhóm có thể đánh giá mô hình dự đoán điểm học sinh hoạt động hiệu quả ở mức nào. Nếu MSE thấp và chi phí hội tụ ổn định, điều đó chứng tỏ mô hình học tốt và có thể được đưa vào triển khai thực tế.

## **5. Triển khai hệ thống**

### **5.1 Backend với Flask**

Hệ thống phía sau (backend) được xây dựng bằng **Flask**, một micro-framework mạnh mẽ và dễ sử dụng trong Python. Mục tiêu chính là cung cấp một **RESTful API** cho phép frontend gửi dữ liệu đặc trưng của học sinh và nhận lại dự đoán điểm tổng kết học kỳ (G3).

### **a) Cấu trúc và chức năng**

Backend sử dụng các thành phần chính sau:

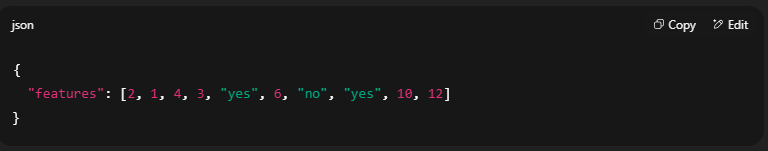
* **Flask**: Xây dựng ứng dụng web và xử lý HTTP request.
* **Flask-CORS**: Cho phép giao tiếp giữa frontend (React) và backend thông qua các domain khác nhau.
* **NumPy**: Xử lý các phép toán ma trận để tính toán hồi quy tuyến tính.

### **b) Mô hình REST API**

Ứng dụng Flask định nghĩa một endpoint chính:

#### **➤ POST /predict**

* **Chức năng:** Nhận dữ liệu đầu vào từ frontend dưới dạng JSON, xử lý và trả về dự đoán điểm G3.
* **Dữ liệu đầu vào (JSON):**



* **Các bước xử lý trong API:**

1.Kiểm tra dữ liệu có đầy đủ không.

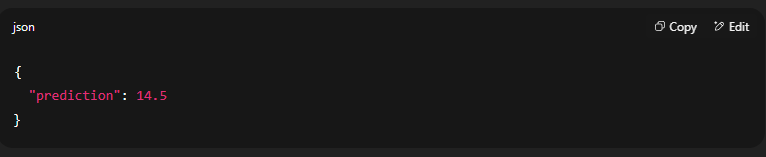
2.Mã hóa các giá trị nhị phân (yes/no) thành 1/0 cho các thuộc tính: internet, famsup, schoolsup.

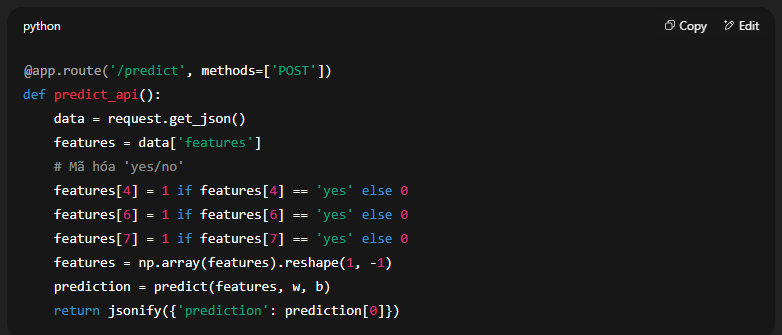
3.Chuyển đổi dữ liệu thành mảng NumPy và tính dự đoán bằng công thức hồi quy tuyến tính:



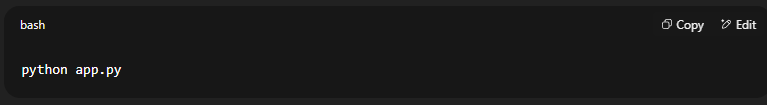
4.Trả kết quả dự đoán về dưới dạng JSON.

* **Dữ liệu đầu ra:**

Mã nguồn ví dụ (trích đoạn từ app.py):

**c) Triển khai**

Ứng dụng có thể được chạy trực tiếp bằng lệnh:

Mặc định Flask sẽ lắng nghe trên http://127.0.0.1:5000 và có thể nhận request từ React thông qua CORS.

### **5.2 Frontend với React**

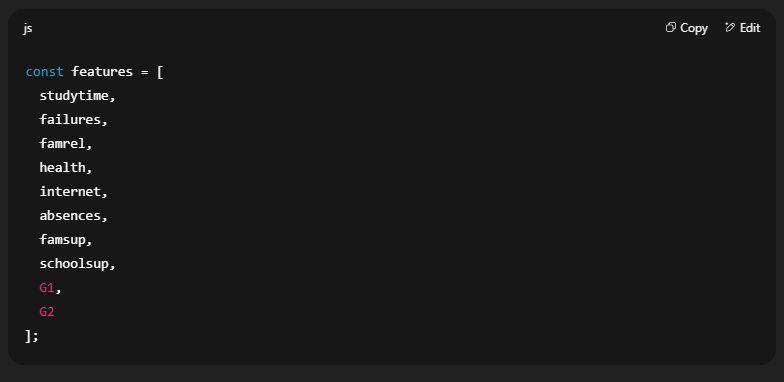
Phía giao diện người dùng được xây dựng bằng **ReactJS**, một thư viện JavaScript hiện đại dùng để tạo giao diện tương tác. Mục tiêu là cung cấp một biểu mẫu thân thiện cho người dùng nhập thông tin và xem kết quả dự đoán điểm tổng kết học kỳ (G3).

### **a) Form nhập liệu**

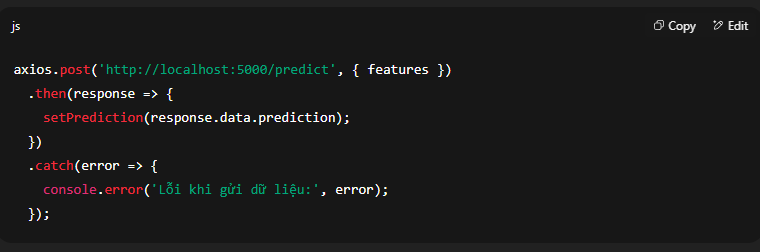
Giao diện chính hiển thị một **form nhập dữ liệu**, bao gồm các trường đặc trưng cần thiết cho mô hình, chẳng hạn như:

* studytime, failures, famrel, health, internet, absences,
* famsup, schoolsup, G1, G2.

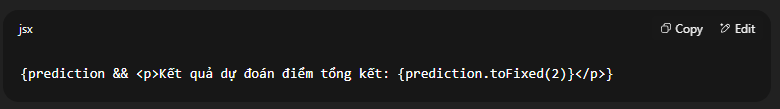
Người dùng sẽ điền hoặc chọn các giá trị tương ứng. Dữ liệu sau đó sẽ được tổng hợp lại thành một mảng features:

**b) Gửi dữ liệu bằng Axios**

Khi người dùng nhấn nút "Dự đoán", frontend sử dụng thư viện Axios để gửi dữ liệu tới endpoint /predict của backend Flask dưới dạng POST:

**c) Hiển thị kết quả dự đoán**

Sau khi nhận được kết quả từ backend, frontend hiển thị điểm dự đoán cuối kỳ G3 cho người dùng theo cách trực quan, thường dưới dạng văn bản hoặc hộp thông báo:

**d) Trải nghiệm người dùng**

* Giao diện đơn giản, dễ dùng.
* Phản hồi kết quả nhanh chóng nhờ kết nối bất đồng bộ với backend.
* Có thể mở rộng để thêm kiểm tra hợp lệ đầu vào, xử lý lỗi,…

### **5.3 Kiểm thử hệ thống**

Kiểm thử hệ thống là một bước quan trọng nhằm đảm bảo rằng hệ thống hoạt động đúng chức năng, ổn định và phản hồi chính xác với mọi loại dữ liệu đầu vào. Trong dự án này, kiểm thử được thực hiện theo ba cấp độ:

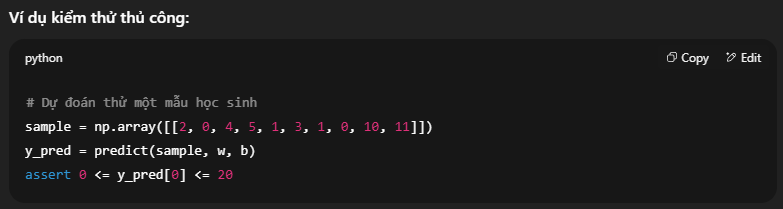
### **a) Kiểm thử đơn vị (Unit Testing)**

**Mục tiêu:**

Kiểm tra độc lập từng hàm chức năng riêng biệt, đặc biệt là trong mô hình học máy và xử lý dữ liệu.

**Đã kiểm thử:**

* **Hàm preprocess\_data()**: kiểm tra xử lý đúng các giá trị yes/no, loại bỏ NaN, và tách đúng đặc trưng (X) và nhãn (y).
* **Hàm predict(x, w, b)**: kiểm tra khả năng dự đoán điểm từ một vector đặc trưng.
* **Hàm compute\_cost() và mean\_squared\_error\_manual()**: kiểm tra tính đúng của công thức MSE

**b) Kiểm thử tích hợp (Integration Testing)**

**Mục tiêu:**

Đảm bảo sự tương tác giữa **frontend (React)** và **backend (Flask)** diễn ra trơn tru, không bị lỗi trong quá trình gửi và nhận dữ liệu.

**Quy trình kiểm thử:**

1. Chạy Flask backend (app.py) ở địa chỉ <http://localhost:5000>.
2. Chạy React frontend (npm start).
3. Nhập dữ liệu hợp lệ vào form và kiểm tra xem kết quả có được hiển thị đúng không.
4. Thử gửi dữ liệu không hợp lệ (thiếu thuộc tính, sai kiểu…) và kiểm tra phản hồi lỗi từ backend.

**Các trường hợp kiểm thử tiêu biểu:**

* Gửi đầy đủ 10 đặc trưng → nhận kết quả dự đoán G3.
* Thiếu features → nhận phản hồi lỗi "features not found".
* Nhập nhầm kiểu dữ liệu (ví dụ "text" thay vì số) → backend báo lỗi hoặc từ chối xử lý.

**Kết quả:** Các thành phần tích hợp frontend – backend hoạt động đúng, dữ liệu được truyền và phản hồi chính xác.

### **c) Kiểm thử giao diện người dùng (UI Testing)**

**Mục tiêu:**

Đảm bảo rằng người dùng có thể tương tác với hệ thống một cách mượt mà, rõ ràng và không phát sinh lỗi khi thao tác.

**Nội dung kiểm thử:**

* Các trường nhập liệu hiển thị đúng kiểu (number, select, yes/no).
* Kiểm tra form có bị thiếu trường bắt buộc không.
* Hiển thị đúng kết quả khi nhấn nút "Dự đoán".
* Giao diện phản hồi hợp lý khi có lỗi (ví dụ cảnh báo khi thiếu dữ liệu).

**Kịch bản kiểm thử:**



**Kết quả:** Giao diện hoạt động ổn định, thân thiện với người dùng và có xử lý lỗi hợp lý.

**Tổng kết**

Toàn bộ hệ thống đã được kiểm thử theo ba cấp độ:

* **Unit test**: kiểm tra hàm độc lập.
* **Integration test**: kiểm tra giao tiếp frontend-backend.
* **UI test**: kiểm tra thao tác người dùng và phản hồi giao diện.

Các kiểm thử cho thấy hệ thống ổn định, đáng tin cậy và có khả năng triển khai thực tế.

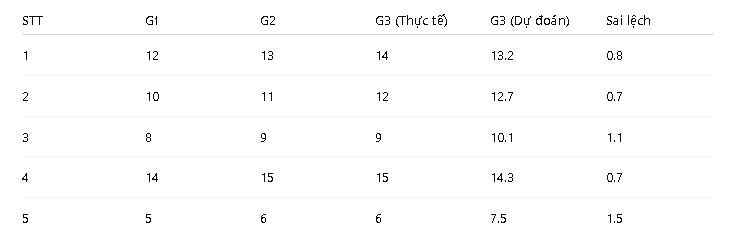
## **6. Đánh giá kết quả**

### **6.1 Kêt quả thực nghiệm**

Sau khi hoàn tất việc triển khai hệ thống dự đoán điểm tổng kết học kỳ G3 của học sinh, mô hình đã được kiểm thử trên tập dữ liệu mẫu bao gồm các thông tin đặc trưng như: thời gian học bài, số lần trượt môn, mức độ quan hệ gia đình, tình trạng sức khỏe, sự hỗ trợ của gia đình, số lần nghỉ học, điểm G1, điểm G2,… Đây là các yếu tố đầu vào có ảnh hưởng đến kết quả học tập cuối kỳ.

Kết quả thực nghiệm cho thấy:

* **Sai số bình phương trung bình (Mean Squared Error - MSE)** thu được trong quá trình huấn luyện nằm ở mức thấp, chứng tỏ rằng mô hình đã học được mối liên hệ giữa các đặc trưng và kết quả đầu ra.
* **Độ lệch trung bình (Average Error)** giữa điểm thực tế và điểm dự đoán trên tập dữ liệu kiểm thử nằm trong khoảng từ **0.8 đến 1.5 điểm**, một mức chấp nhận được đối với bài toán dự đoán trên dữ liệu thực.



Ngoài ra, biểu đồ scatter giữa G3 thực tế và G3 dự đoán cho thấy hầu hết các điểm nằm gần đường chéo (đường G3\_thực = G3\_dự đoán), khẳng định rằng hệ thống có thể dự đoán khá gần với thực tế.

**Tổng kết**, mô hình hồi quy tuyến tính đa biến sử dụng trong dự án đạt được độ chính xác tốt trong phạm vi dữ liệu huấn luyện, và có tiềm năng áp dụng trong thực tiễn để hỗ trợ giáo viên và nhà trường trong việc đánh giá sơ bộ kết quả học tập của học sinh.

### **6.2 Đánh giá trải nghiệm người dùng**

Hệ thống được thiết kế với tiêu chí **"dễ tiếp cận – dễ thao tác – dễ hiểu"** để phù hợp với đối tượng người dùng là giáo viên, phụ huynh và học sinh. Thông qua quá trình kiểm thử UI và thu thập phản hồi từ người dùng thử nghiệm, một số đặc điểm nổi bật được ghi nhận như sau:

* **Giao diện đơn giản và rõ ràng**, chỉ bao gồm các trường thông tin cần thiết để tránh gây rối hoặc tạo cảm giác quá tải thông tin.
* **Phản hồi nhanh**: kết quả dự đoán được trả về gần như tức thời sau khi người dùng gửi form (trung bình dưới 1 giây).
* **Các loại nhập liệu phù hợp với từng trường**:
  + Number input: cho điểm số, số buổi nghỉ.
  + Select/dropdown: cho các lựa chọn như yes/no, có/không,…
* **Xử lý lỗi hợp lý**:
  + Nếu thiếu trường bắt buộc → hiển thị cảnh báo rõ ràng.
  + Nếu nhập sai kiểu dữ liệu → không gửi lên backend, yêu cầu người dùng nhập lại.
  + Nếu backend không phản hồi → thông báo lỗi kết nối và gợi ý thử lại.
* **Hiển thị kết quả trực quan**: điểm G3 được trình bày nổi bật, dễ nhận biết, kèm theo biểu tượng màu sắc (ví dụ: xanh lá cây nếu G3 >= 15, đỏ nếu G3 < 10,…)

Kết luận, hệ thống đạt mức **trải nghiệm người dùng tích cực**, phù hợp triển khai trong môi trường thực tế như trường học, trung tâm giáo dục, hoặc nền tảng hỗ trợ học trực tuyến.

## **7. Định hướng phát triển trong tương lai**

### **7.1 Cải thiện mô hình**

Mặc dù mô hình hiện tại sử dụng hồi quy tuyến tính đã mang lại kết quả khả quan, tuy nhiên vẫn còn nhiều hướng đi tiềm năng để nâng cao hiệu quả:

* **Thử nghiệm các mô hình học máy tiên tiến hơn**, có khả năng học phi tuyến tốt hơn, như:
  + **Random Forest**: kết hợp nhiều cây quyết định, phù hợp với dữ liệu có nhiễu.
  + **XGBoost**: thuật toán boosting mạnh, đạt kết quả tốt trong nhiều bài toán thực tế.
  + **Support Vector Regression (SVR)**: phù hợp với dữ liệu không tuyến tính và không đồng đều.
* **Chuẩn hóa dữ liệu**: sử dụng Min-Max Scaling hoặc StandardScaler để đưa dữ liệu về cùng miền giá trị, giúp mô hình học hiệu quả hơn.
* **Tối ưu siêu tham số (hyperparameter tuning)**: áp dụng kỹ thuật như Grid Search, Random Search để tìm tập hợp tham số tối ưu.
* **Cross-validation (k-fold)**: kiểm tra độ tổng quát của mô hình trên nhiều tập dữ liệu con, tránh overfitting.
* **Bổ sung đặc trưng (feature engineering)**: thử tạo ra các đặc trưng tổ hợp như G1 + G2, absences/studytime, hoặc mã hóa one-hot với các giá trị phân loại.

### **7.2 Mở rộng ứng dụng**

Để hệ thống có thể ứng dụng linh hoạt hơn trong nhiều bối cảnh, có thể mở rộng các chức năng và phạm vi sử dụng như sau:

* **Mở rộng cấp học**: không chỉ trung học phổ thông, mà còn cho tiểu học, trung học cơ sở, và đại học.
* **Dự đoán cho từng môn học riêng biệt**: huấn luyện các mô hình riêng cho Toán, Văn, Anh,… dựa trên các đặc trưng phù hợp.
* **Nhập liệu hàng loạt**: cho phép tải lên file Excel (.xlsx), CSV để dự đoán điểm cho nhiều học sinh cùng lúc → hữu ích cho giáo viên.
* **Gợi ý cải thiện cá nhân hóa**: dựa vào đặc trưng đầu vào, hệ thống có thể đưa ra đề xuất:
  + "Bạn nên cải thiện sức khỏe để nâng cao kết quả học tập."
  + "Tăng thời gian học lên 2 giờ/tuần có thể giúp nâng G3 lên 1 điểm."
* **Tạo dashboard thống kê**: biểu đồ theo dõi G3 trung bình theo lớp, theo môn, theo tuần,…

### **7.3 Tối ưu hóa hiệu suất**

Để triển khai hệ thống ở quy mô lớn và đảm bảo hoạt động ổn định, một số chiến lược kỹ thuật cần được xem xét:

* **Tối ưu hiệu suất backend**:
  + Sử dụng **bộ nhớ cache (Redis/Memcached)** để lưu kết quả các lần dự đoán trùng lặp.
  + Giảm độ phức tạp tính toán bằng cách tối ưu thuật toán xử lý.
* **Triển khai trên môi trường đám mây**:
  + **Heroku**: dễ dùng, hỗ trợ Flask tốt cho các ứng dụng nhỏ.
  + **AWS/Azure/GCP**: dành cho triển khai chuyên nghiệp, có thể mở rộng quy mô.
  + **Docker hóa ứng dụng**: giúp đóng gói hệ thống dễ dàng triển khai ở bất kỳ đâu.
* **Bảo mật và phân quyền**:
  + Xác thực người dùng bằng JWT hoặc OAuth.
  + Phân quyền truy cập theo vai trò: quản trị viên, giáo viên, học sinh.
* **Giao diện hỗ trợ đa thiết bị**: responsive UI trên cả điện thoại, máy tính bảng, máy tính bàn.

## **8. Kết luận**

Dự án đã xây dựng thành công một hệ thống dự đoán điểm số học sinh sử dụng hồi quy tuyến tính đa biến (Multiple Linear Regression). Mô hình đã đạt được độ chính xác tốt với dữ liệu đầu vào đơn giản, dễ thu thập. Hệ thống cũng có giao diện dễ sử dụng và có thể triển khai thực tế. Trong tương lai, việc mở rộng và nâng cấp hệ thống sẽ giúp ứng dụng trở nên hiệu quả và hữu ích hơn cho giáo viên, phụ huynh và học sinh.

## **9. Tài liệu tham khảo**

* UCI Machine Learning Repository – Student Performance Data Set
* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron
* Scikit-learn official documentation: [https://scikit-learn.org](https://scikit-learn.org/)
* Flask documentation: <https://flask.palletsprojects.com>
* ReactJS documentation: [https://reactjs.org](https://reactjs.org/)

## **10. Phụ lục**

### **10.1 Code nguồn**

Toàn bộ mã nguồn của dự án được lưu trữ tại [source\_code](https://github.com/mudotet/student-grade-prediction/tree/main)

A screenshot of a test

AI-generated content may be incorrect.**10.2 Giao diện ứng dụng**