TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

——————— o0o ———————

BÁO CÁO GIỮA KỲ

PHƯƠNG PHÁP SỐ CHO HỌC MÁY

Các mô hình hồi quy tuyến tính được sử dụng nhiều trong Học có giám sát (Surpervised Learning). Trình bày các bài toán tối ưu liên quan và các phương pháp giải sau: least squared estimation và Shrinkage methods (Ridge regression, lasso regression, spase regression)

Lecturer: PhD. Nguyễn Trung Thành

Class: Phương Pháp Số Cho Học Máy-1-2-23(N01)

Tên sinh viên	**MSSV**
Đào Đức Minh	21010555
Lê Vĩnh Hưng	21011494

Hà Nội, 2024

Mục lục

Bảng phân công công việc

Tên sinh viên	**Công việc đã làm**
Đào Đức Minh	Ridge, LASSO Regression và chạy thực nghiệm
Lê Vĩnh Hưng	Least Square Estimation, tìm dataset và tiền xử lý dữ liệu, viết báo cáo

Tóm tắt nội dung

Các mô hình hồi quy tuyến tính là một loại thuật toán học có giám sát phổ biến nhằm tìm mối quan hệ tuyến tính giữa một tập các biến đầu vào và một biến mục tiêu.

Tuy nhiên, các mô hình hồi quy tuyến tính có thể gặp phải các vấn đề như quá khớp, đa cộng tuyến, hoặc phương sai cao khi số lượng biến đầu vào lớn hoặc dữ liệu nhiễu. Để khắc phục những thách thức này, các phương pháp tối ưu hóa khác nhau đã được đề xuất để cải thiện hiệu năng và ổn định của các mô hình hồi quy tuyến tính. Trong dự án này, chúng tôi trình bày các bài toán tối ưu hóa liên quan và các phương pháp giải quyết sau: ước lượng bình phương tối thiểu và các phương pháp thu gọn (hồi quy ridge, hồi quy lasso, và hồi quy thưa).

Chúng tôi so sánh và đối chiếu các phương pháp này về các giả định, ưu điểm, nhược điểm, và ứng dụng của chúng. Chúng tôi cũng triển khai các phương pháp này trên một số tập dữ liệu nhân tạo và thực tế và đánh giá kết quả của chúng bằng các chỉ số phù hợp.

Keywords: Optimization Algorithms - Machine Learning

Chương 1

Giới thiệu chung

1.1 Lịch sử về thuật toán tối ưu

Các thuật toán tối ưu hóa là những phương pháp để tìm ra giải pháp tốt nhất hoặc tối ưu cho một bài toán, chẳng hạn như tối thiểu hóa một hàm chi phí hoặc tối đa hóa một hàm tiện ích. Các thuật toán tối ưu hóa là thiết yếu cho học máy, vì hầu hết các thuật toán học máy đều liên quan đến việc xây dựng một mô hình tối ưu hóa và học các tham số từ dữ liệu.

Lịch sử của các thuật toán tối ưu hóa có thể truy nguyên về thời cổ đại, khi con người cố gắng giải quyết các bài toán như tìm đường đi ngắn nhất, chia đất, hoặc phân bổ tài nguyên. Tuy nhiên, kỷ nguyên hiện đại của các thuật toán tối ưu hóa bắt đầu với sự phát triển của giải tích và đại số tuyến tính, cho phép đặt ra và giải quyết các bài toán tuyến tính, phi tuyến, và biến phân.

Trong thế kỷ hai mươi, các thuật toán tối ưu hóa trở nên đa dạng và tinh vi hơn, khi các lĩnh vực mới của toán học và khoa học máy tính xuất hiện. Một số thuật toán tối ưu hóa có ảnh hưởng trong giai đoạn này bao gồm phương pháp đơn hình, phương pháp gradient xuống, phương pháp Newton-Raphson, điều kiện Kuhn-Tucker, điều kiện Karush-Kuhn-Tucker, phương pháp nhánh và cận, phương pháp quy hoạch động, phương pháp quy hoạch tuyến tính, phương pháp quy hoạch bậc hai, phương pháp điểm trong, thuật toán di truyền, phương pháp nhiệt độ giả, phương pháp tìm kiếm tabu, phương pháp tối ưu hóa bầy kiến, phương pháp tối ưu hóa bầy hạt, và phương pháp mạng nơ-ron nhân tạo.

Ngày nay, các thuật toán tối ưu hóa được sử dụng rộng rãi trong các lĩnh vực khác nhau

của học máy, như học có giám sát, học không giám sát, học tăng cường, học sâu, và học Bayes. Các thuật toán tối ưu hóa liên tục phát triển và cải tiến, khi các bài toán, mô hình, và kỹ thuật mới xuất hiện. Một số xu hướng và thách thức hiện tại trong các thuật toán tối ưu hóa cho học máy bao gồm việc xử lý các bài toán quy mô lớn, phức tạp, nhiễu, động, và đa mục tiêu, cũng như kết hợp các loại thuật toán tối ưu hóa khác nhau và học từ dữ liệu.

1.2 Vấn đề bài toán

Học có giám sát là một loại học máy mà trong đó việc học một hàm ánh xạ một tập các biến đầu vào $\mathbf{x}$ tới một biến mục tiêu y, dựa trên một tập dữ liệu huấn luyện cho trước $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Các mô hình hồi quy tuyến tính là một loại thuật toán học có giám sát phổ biến mà giả định một mối quan hệ tuyến tính giữa $\mathbf{x}$ và y, sao cho $y = \mathbf{w}^T\mathbf{x} + b$, trong đó $\mathbf{w}$ là một vector trọng số và b là một hệ số chệch.

Mục tiêu của các mô hình hồi quy tuyến tính là tìm ra các giá trị tối ưu của $\mathbf{w}$ và b sao cho tối thiểu hóa một hàm chi phí, mà đo lường sự sai khác giữa các giá trị dự đoán và các giá trị thực của y. Một lựa chọn phổ biến của hàm chi phí là trung bình bình phương sai số (MSE), được định nghĩa như sau:

$$\text{MSE}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i - b)^2$$

Bài toán tối ưu hóa của các mô hình hồi quy tuyến tính có thể được viết như sau:

$$\min_{\mathbf{w},b} \text{MSE}(\mathbf{w}, b)$$

Đây là một bài toán tối ưu hóa lồi mà có thể được giải một cách phân tích bằng phương trình bình thường hoặc một cách số học bằng các phương pháp dựa trên gradient. Tuy nhiên, các mô hình hồi quy tuyến tính có thể gặp phải các vấn đề như quá khớp, đa cộng tuyến, hoặc phương sai cao khi số lượng biến đầu vào lớn hoặc dữ liệu nhiễu. Để khắc phục những thách thức này, các phương pháp điều chuẩn khác nhau đã được đề xuất để cải thiện hiệu năng và ổn định của các mô hình hồi quy tuyến tính. Các phương pháp điều chuẩn thêm một hạng phạt vào hàm chi phí, mà thu gọn các giá trị của các trọng số về không hoặc một số giá trị khác. Điều này giảm độ phức tạp của mô hình và ngăn chặn quá khớp.

Trong phần này, chúng tôi sẽ trình bày các bài toán tối ưu hóa liên quan và các phương pháp điều chuẩn sau: ước lượng bình phương tối thiểu và các phương pháp thu gọn (hồi quy ridge, hồi quy lasso, và hồi quy thưa). Chúng tôi sẽ so sánh và đối chiếu các phương pháp này về các giả định, ưu điểm, nhược điểm, và ứng dụng của chúng. Chúng tôi cũng triển khai các phương pháp này trên một số tập dữ liệu nhân tạo và thực tế và đánh giá kết quả của chúng bằng các chỉ số phù hợp.

Chương 2

Mô hình tối ưu và ước lượng

Cách ước lượng, tối ưu, và so sánh các phương pháp hồi quy dựa trên nhiều yếu tố:

- Hàm mục tiêu: là hàm số biểu diễn mục đích của bài toán, thường là tối thiểu hóa sai số giữa giá trị dự đoán và giá trị thực tế của biến mục tiêu. Một số ví dụ của hàm mục tiêu là sai số bình phương trung bình (MSE), sai số tuyệt đối trung bình (MAE), sai số bình phương nhỏ nhất (OLS), v.v.

- Phương pháp ước lượng: là phương pháp để tìm ra các tham số của mô hình hồi quy, sao cho hàm mục tiêu đạt giá trị tối ưu. Một số ví dụ của phương pháp ước lượng là phương trình bình thường, phương pháp gradient, phương pháp Newton-Raphson, phương pháp bình phương tối thiểu, v.v.

- Phương pháp điều chuẩn: là phương pháp để giảm độ phức tạp của mô hình hồi quy, bằng cách thêm một hạng phạt vào hàm mục tiêu, để tránh hiện tượng quá khớp (overfitting) hoặc đa cộng tuyến (multicollinearity). Một số ví dụ của phương pháp điều chuẩn là hồi quy ridge, hồi quy lasso, hồi quy elastic net, v.v.

- Phương pháp đánh giá: là phương pháp để đo lường hiệu năng của mô hình hồi quy, bằng cách sử dụng các chỉ số thống kê hoặc đồ thị để so sánh giữa giá trị dự đoán và giá trị thực tế của biến mục tiêu. Một số ví dụ của phương pháp đánh giá là hệ số xác định (R-squared), sai số chuẩn (RMSE), sai số tuyệt đối trung bình phần trăm (MAPE), đồ thị phần dư (residual plot), v.v.

Ở phần này, chúng ta sẽ liệt kê và giải các phương pháp tối ưu được đưa ra trong đề tài của báo cáo. Tiếp theo đó là giới thiệu các thước đo và cách ước lượng các phương pháp này.

2.1 Các phương pháp tối ưu

2.1.1 Least Square Estimation

Bài toán least square estimation là một bài toán tối ưu hóa, trong đó hàm mục tiêu là tối thiểu hóa tổng bình phương sai số giữa các giá trị dự đoán và các giá trị thực tế của một biến phụ thuộc, dựa trên một hoặc nhiều biến độc lập. Bài toán này thường được sử dụng để khớp một mô hình tuyến tính hoặc phi tuyến với dữ liệu quan sát được.

Hàm mục tiêu của bài toán least square estimation:

$$\min_{\mathbf{w},b} \text{SSE}(\mathbf{w}, b) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i - b)^2$$

Trong đó $\mathbf{w}$ là vector trọng số, b là hệ số chệch, $\mathbf{x}_i$ là vector đầu vào, y_i là giá trị mục tiêu, và n là số lượng dữ liệu.

Các ràng buộc của bài toán least square estimation:

$$\mathbf{w}, b \in R$$

Cách giải bài toán least square estimation:

- Giải phân tích bằng phương trình bình thường:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

$$b = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)$$

Trong đó $\mathbf{X}$ là ma trận đầu vào, $\mathbf{y}$ là vector mục tiêu.

- Giải số học bằng các phương pháp dựa trên gradient, như gradient xuống, gradient ngẫu nhiên, hoặc gradient hạt nhân. Các phương pháp này cập nhật $\mathbf{w}$ và b theo công thức:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial \text{SSE}}{\partial \mathbf{w}}$$

$$b \leftarrow b - \alpha \frac{\partial \text{SSE}}{\partial b}$$

Trong đó α là tốc độ học, $\frac{\partial \text{SSE}}{\partial \mathbf{w}}$ và $\frac{\partial \text{SSE}}{\partial b}$ là đạo hàm riêng của hàm SSE theo $\mathbf{w}$ và b.

2.1.2 Ridge Regression

Ridge Regression là một phương pháp điều chuẩn cho bài toán hồi quy tuyến tính, mà mục tiêu là tìm ra các tham số sao cho tối thiểu hóa sai số bình phương trung bình (MSE) giữa các giá trị dự đoán và các giá trị thực tế, đồng thời giảm độ lớn của các tham số bằng cách thêm một hạng phạt vào hàm mục tiêu.

Hàm mục tiêu của bài toán Ridge Regression:

$$\min_{\mathbf{w},b} \text{MSE}(\mathbf{w}, b) + \lambda \|\mathbf{w}\|^2$$

Trong đó $\mathbf{w}$ là vector trọng số, b là hệ số chệch, $\mathbf{x}_i$ là vector đầu vào, y_i là giá trị mục tiêu, n là số lượng dữ liệu, và λ là tham số điều chuẩn.

Các ràng buộc của bài toán Ridge Regression:

$$\mathbf{w}, b \in R$$

$$\lambda \geq 0$$

Cách giải bài toán Ridge Regression:

- Giải phân tích bằng phương trình bình thường:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

$$b = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i)$$

Trong đó $\mathbf{X}$ là ma trận đầu vào, $\mathbf{y}$ là vector mục tiêu, và $\mathbf{I}$ là ma trận đơn vị.

- Giải số học bằng các phương pháp dựa trên gradient, như gradient xuống, gradient ngẫu nhiên, hoặc gradient hạt nhân. Các phương pháp này cập nhật $\mathbf{w}$ và b theo công thức:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\frac{\partial \text{MSE}}{\partial \mathbf{w}} + 2\lambda \mathbf{w})$$

$$b \leftarrow b - \alpha \frac{\partial \text{MSE}}{\partial b}$$

Trong đó α là tốc độ học, $\frac{\partial \text{MSE}}{\partial \mathbf{w}}$ và $\frac{\partial \text{MSE}}{\partial b}$ là đạo hàm riêng của hàm MSE theo $\mathbf{w}$ và b.

2.1.3 Lasso Regression

Lasso Regression là một phương pháp điều chuẩn cho bài toán hồi quy tuyến tính, mà mục tiêu là tìm ra các tham số sao cho tối thiểu hóa sai số bình phương trung bình (MSE) giữa các giá trị dự đoán và các giá trị thực tế, đồng thời giảm độ lớn của các tham số bằng cách thêm một hạng phạt vào hàm mục tiêu.

Hàm mục tiêu của bài toán Lasso Regression:

$$\min_{\mathbf{w},b} \text{MSE}(\mathbf{w}, b) + \lambda \|\mathbf{w}\|_1$$

Trong đó $\mathbf{w}$ là vector trọng số, b là hệ số chệch, $\mathbf{x}_i$ là vector đầu vào, y_i là giá trị mục tiêu, n là số lượng dữ liệu, λ là tham số điều chuẩn, và $\|\mathbf{w}\|_1$ là chuẩn 1 của $\mathbf{w}$, được định nghĩa là:

$$\|\mathbf{w}\|_1 = \sum_{i=1}^{n} |w_i|$$

Các ràng buộc của bài toán Lasso Regression:

$$\mathbf{w}, b \in R$$

$$\lambda \geq 0$$

Cách giải bài toán Lasso Regression:

- Giải số học bằng các phương pháp dựa trên gradient, như gradient xuống, gradient ngẫu nhiên, hoặc gradient hạt nhân. Các phương pháp này cập nhật $\mathbf{w}$ và b theo công thức:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\frac{\partial \text{MSE}}{\partial \mathbf{w}} + \lambda \text{sign}(\mathbf{w}))$$

$$b \leftarrow b - \alpha \frac{\partial \text{MSE}}{\partial b}$$

Trong đó α là tốc độ học, $\frac{\partial \text{MSE}}{\partial \mathbf{w}}$ và $\frac{\partial \text{MSE}}{\partial b}$ là đạo hàm riêng của hàm MSE theo $\mathbf{w}$ và b, và $\text{sign}(\mathbf{w})$ là hàm dấu của $\mathbf{w}$, được định nghĩa là:

$$\text{sign}(\mathbf{w}) = \begin{cases} 1 & \text{if } \mathbf{w} > 0 \\ 0 & \text{if } \mathbf{w} = 0 \\ -1 & \text{if } \mathbf{w} < 0 \end{cases}$$

So sánh giữa Lasso và Ridge

Khác nhau giữa Lasso và ridge Regression:

- Lasso regression sử dụng chuẩn 1 của vector trọng số làm hạng phạt, trong khi ridge regression sử dụng chuẩn 2 bình phương của vector trọng số làm hạng phạt.
- Lasso regression có thể loại bỏ các tham số không quan trọng bằng cách đặt chúng bằng 0, trong khi ridge regression chỉ có thể thu gọn chúng về 0 nhưng không bao giờ bằng 0.
- Lasso regression có thể xử lý tốt hơn các trường hợp có nhiều tính năng có độ tương quan cao, trong khi ridge regression có thể xử lý tốt hơn các trường hợp có số lượng đối tượng lớn hơn số lượng quan sát và nhiều đối tượng có đa cộng tuyến.

Vậy sự khác biệt của 2 phương pháp này so với phương pháp đầu tiên nằm ở chỗ:

- Least square estimation không sử dụng hạng phạt nào, nên có thể dẫn đến hiện tượng quá khớp (overfitting) hoặc đa cộng tuyến (multicollinearity) nếu số lượng biến độc lập lớn hơn số lượng quan sát hoặc có mối tương quan cao với nhau.

2.1.4 ElasticNet Regression

ElasticNet Regression là một phương pháp điều chuẩn cho bài toán hồi quy tuyến tính, mà mục tiêu là tìm ra các tham số sao cho tối thiểu hóa sai số bình phương trung bình (MSE) giữa các giá trị dự đoán và các giá trị thực tế, đồng thời giảm độ lớn của các tham số bằng cách thêm một hạng phạt vào hàm mục tiêu.

Hàm mục tiêu của bài toán ElasticNet Regression:

$$\min_{\mathbf{w},b} \text{MSE}(\mathbf{w}, b) + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|^2$$

Trong đó $\mathbf{w}$ là vector trọng số, b là hệ số chệch, $\mathbf{x}_i$ là vector đầu vào, y_i là giá trị mục tiêu, n là số lượng dữ liệu, λ_1 và λ_2 là các tham số điều chuẩn, $\|\mathbf{w}\|_1$ là chuẩn 1 của $\mathbf{w}$, và $\|\mathbf{w}\|^2$ là chuẩn 2 bình phương của $\mathbf{w}$.

Các ràng buộc của bài toán ElasticNet Regression:

$$\mathbf{w}, b \in R$$

$$\lambda_1, \lambda_2 \geq 0$$

Cách giải bài toán ElasticNet Regression:

- Giải số học bằng các phương pháp dựa trên gradient, như gradient xuống, gradient ngẫu nhiên, hoặc gradient hạt nhân. Các phương pháp này cập nhật $\mathbf{w}$ và b theo công thức:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\frac{\partial \text{MSE}}{\partial \mathbf{w}} + \lambda_1 \text{sign}(\mathbf{w}) + 2\lambda_2 \mathbf{w})$$

$$b \leftarrow b - \alpha \frac{\partial \text{MSE}}{\partial b}$$

Trong đó α là tốc độ học, $\frac{\partial \text{MSE}}{\partial \mathbf{w}}$ và $\frac{\partial \text{MSE}}{\partial b}$ là đạo hàm riêng của hàm MSE theo $\mathbf{w}$ và b, và $\text{sign}(\mathbf{w})$ là hàm dấu của $\mathbf{w}$.

Lasso, Ridge và ElasticNet là các phương pháp điều chuẩn cho bài toán hồi quy tuyến tính, mà mục tiêu là tìm ra các tham số sao cho tối thiểu hóa sai số bình phương trung bình (MSE) giữa các giá trị dự đoán và các giá trị thực tế, đồng thời giảm độ lớn của các tham số bằng cách thêm một hạng phạt vào hàm mục tiêu.

So sánh với các phương pháp còn lại

Sự khác biệt giữa ElasticNet so với 3 phương pháp còn lại:

- ElasticNet là sự kết hợp hoàn hảo giữa Lasso và Ridge bằng cách thêm cả hai hạng phạt chuẩn 1 và chuẩn 2 vào hàm mục tiêu, có thể giảm độ phức tạp và tăng độ chính xác của mô hình hồi quy.

2.2 R2 score

R2 là chỉ số thường được sử dụng để đánh giá mô hình hồi quy, bằng cách so sánh giữa các giá trị dự đoán và các giá trị thực tế của biến mục tiêu.

R2 (hệ số xác định) là tỷ lệ phần trăm của biến thiên của biến mục tiêu được giải thích bởi các biến độc lập. R2 càng gần 1 thì mô hình càng phù hợp với dữ liệu, và ngược lại. Công thức tính R2 là:

$$R^2 = 1 - \frac{\text{SSE}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Trong đó SSE là tổng bình phương sai số, TSS là tổng bình phương toàn bộ, y_i là giá trị thực tế, $\hat{y}_i$ là giá trị dự đoán, và $\bar{y}$ là giá trị trung bình của biến mục tiêu.

Đối với R2, ta có thể so sánh trực tiếp giữa các mô hình khác nhau, bởi vì R2 không phụ thuộc vào đơn vị của biến mục tiêu. Mô hình nào có R2 cao hơn thì có nghĩa là phù hợp hơn với dữ liệu.

Còn đối với MSE, ta chỉ có thể so sánh giữa các mô hình cùng có biến mục tiêu với đơn vị như nhau, bởi vì MSE có cùng đơn vị với bình phương của biến mục tiêu. Mô hình nào có MSE thấp hơn thì có nghĩa là chính xác hơn.

Tương quan của R2 score đối với mô hình

Trong ngữ cảnh của điều chuẩn hồi quy tuyến tính (như Ridge, Lasso và ElasticNet), tham số alpha được dùng để chỉ độ mạnh của điều chuẩn, alpha càng lớn, mức độ điều chuẩn càng mạnh, và ngược lại:

- Đối với Ridge (L2 regularition), alpha là hệ số nhân với tổng bình phương của các hệ số hồi quy trong hàm mất mát. Thật vậy, L2 chỉ giảm trọng số nhưng không làm chúng bằng không, giúp tăng tính tổng quát hoá của mô hình.

- Đối với Lasso (L1 regularition), alpha là hệ số nhân với tổng giá trị tuyệt đối của các hệ số hồi quy trong hàm mất mát. Điều này có thể dẫn đến một số trọng số bằng 0, giúp việc thực hiện chọn lọc feature.

- Đối với ElasticNet có cả hai chức năng của Ridge và Lasso nên cũng có thể nói đây là mô hình được nâng cấp từ hai phương pháp kể trên. Nhưng cũng không vì thế mà phương pháp tối ưu này hiệu quả trên mọi data đưa vào mô hình.

Nhìn chung, để đánh giá được hàm tối ưu trên một mô hình nào đó với dataset nhất định, ta phải thực nghiệm từng mô hình với từng phương pháp và so sánh các chỉ số lân cận. Như phần thực nghiệm bên dưới, ta sẽ đánh giá thông qua 3 thước đo: MSE, R2 score và thời gian chạy mô hình để có được kết quả trực quan nhất.

Chương 3

Thực nghiệm

Ở chương này, báo cáo sẽ đưa ra 2 tập dataset được dùng xuyên suốt quá trình thực nghiệm. Cách xử lý data và số liệu thống kê sau khi huấn luyện cũng như so sánh chúng với nhau.

3.1 Dataset

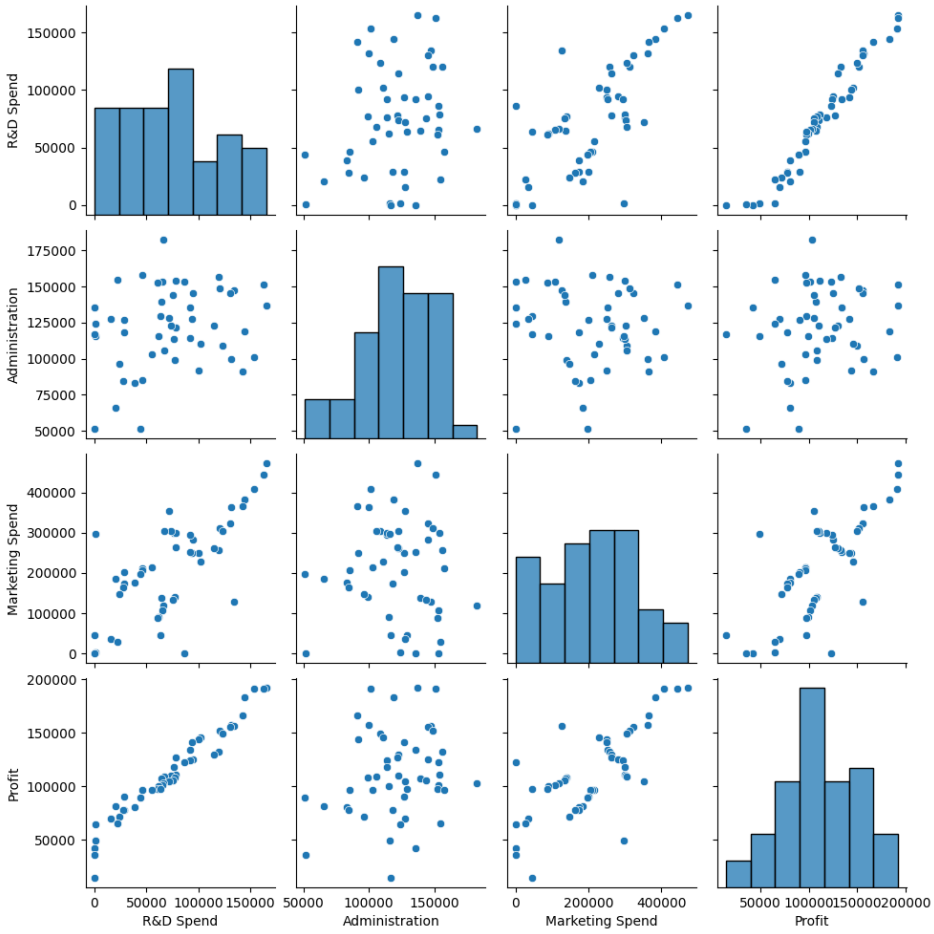
- 50 Startups: Get link
- Epicurious - Recipes with Rating and Nutrition: Get link

Hai tập data trên đều là ở dạng bảng. Trong đó tập 50 Startups nhỏ hơn. Chi tiết về mỗi loại data sẽ được trình bày ở phần dưới.

3.1.1 50 Startups

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

Tập data 50 Startup là một tập dữ liệu chứa thông tin về 50 công ty khởi nghiệp, bao gồm chi phí nghiên cứu và phát triển (R&D Spend), chi phí quản lý (Administration), chi



phí tiếp thị (Marketing Spend), lợi nhuận (Profit) và bang hoạt động (State) của các công ty đó. Tập dữ liệu có 50 hàng và 5 cột, không có giá trị bị thiếu.

- bao gồm tất cả 5 features và 50 objects

Bài toán cần giải của tập 50 startup là bài toán hồi quy tuyến tính đa biến, mà mục tiêu là dự đoán lợi nhuận của các công ty khởi nghiệp dựa trên các biến đầu vào là chi phí nghiên cứu và phát triển, chi phí quản lý, chi phí tiếp thị và bang hoạt động.

Xử lý data

Do tập data này chỉ có 3 cột giá trị dùng để học và một cột dùng làm giá trị thực nên xử lý rất đơn giản là đọc data sau đó đưa cột 1,2,3 làm X và cột ngoài cùng(5) sẽ làm y, ta được X = [50, 3] và y = [50, 1]:

```
X = data.iloc[:,:3].values
y = data.iloc[:,4:5].values
print(X.shape, y.shape)
```

```
(50, 3) (50, 1)
```

3.1.2 Epicurious - Recipes with Rating and Nutrition

	title	rating	calories	protein	fat	sodium	#cakeweek	#wasteless	22-minute meals	3-ingredient recipes	...	yellow squash	yogurt	yonkers	yuca	zucchini	cookbooks	leftovers	snack	snack week	turkey
0	Lentil, Apple, and Turkey Wrap	2.500	426.0	30.0	7.0	559.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1	Boudin Blanc Terrine with Red Onion Confit	4.375	403.0	18.0	23.0	1439.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	Potato and Fennel Soup Hodge	3.750	165.0	6.0	7.0	165.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Spinach Noodle Casserole	3.125	547.0	20.0	32.0	452.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	The Best Blts	4.375	948.0	19.0	79.0	1042.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tập dữ liệu này chứa các công thức nấu ăn từ trang web Epicurious, bao gồm các thông tin:

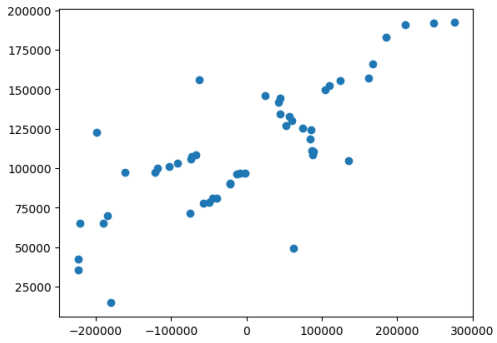
- **Đánh giá**: Điểm số từ 0 đến 4 cho mỗi công thức, dựa trên lượt bình chọn của người dùng.
- **Dinh dưỡng**: Lượng calo, protein, chất béo và natri trong mỗi khẩu phần, tính bằng gam.
- **Thể loại**: Các nhãn phân loại công thức theo loại món ăn, nguyên liệu, mùa, dịp lễ, v.v.
- **Tiêu đề**: Tên của công thức, thường bao gồm tên món chính và một số thành phần chính.
- **Mô tả**: Một đoạn văn ngắn giới thiệu công thức, thường bao gồm nguồn gốc, cách chế biến và một số mẹo.

Dùng PCA để giảm chiều dữ liệu

```
pca = PCA(n_components=34)   # Khởi tạo PCA với 34 thành phần chính
pca.fit(X)  # Fit PCA với dữ liệu
X_pca = pca.transform(X)    # Chuyển đổi dữ liệu sang thành phần chính đầu tiên
print(X_pca.shape, y.shape)
```

✓ 0.5s

```
(15864, 34) (15864,)
```



- .....
- bao gồm tất cả 680 features và 15864 objects

Tập dữ liệu này có thể được sử dụng cho các mục đích khác nhau, chẳng hạn như phân tích xu hướng ẩm thực, đề xuất công thức, phát hiện thành phần, v.v.

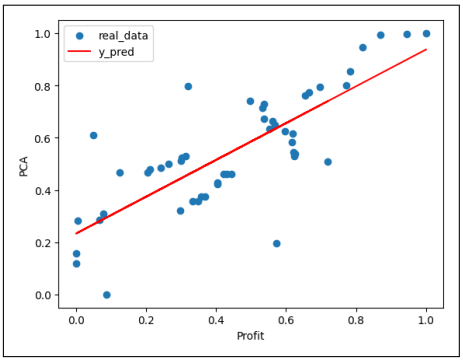
Do tập data này mang một số lượng lớn features và đều là giá trị float không bao gồm null nên chúng tôi chỉ chọn các cột nguyên liệu được thể hiện dưới dạng onehot encoded và cột "calories" để làm dữ liệu training cũng như đã sử dụng phương pháp PCA để giảm chiều dữ liệu xuống còn 34 và 10 features.

3.2 Mô hình hồi quy

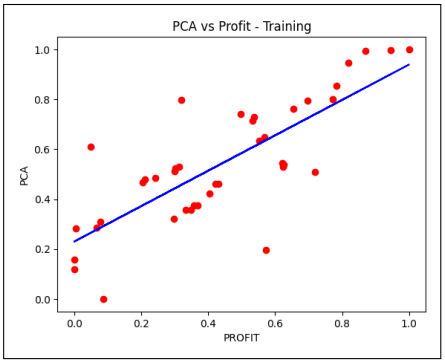
Ứng với 2 dataset đã cho, sẽ có 2 bài toán: Dự đoán trend và phân loại ứng với 2 mô hình

3.2.1 Tối ưu mô hình Linear Regression

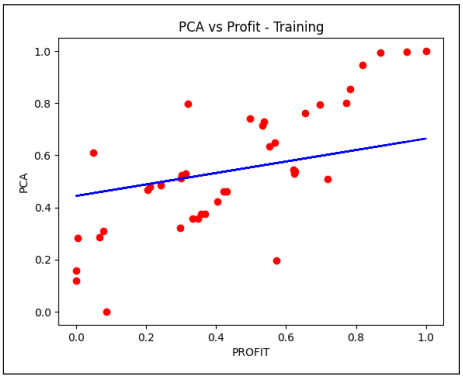
Khởi tạo mô hình, ta áp dụng từng phương pháp vào hàm loss để minimize nó. Vậy sẽ có 4 mô hình trong đó phương pháp MSE được dùng như một thước đo để tính các phương pháp còn lại. Một vài kết quả chạy bài toán:



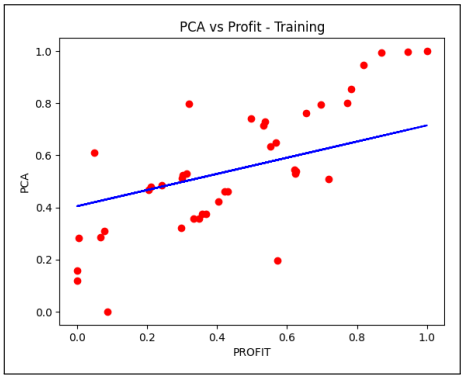
Least Square Estimation



Ridge Regression



Lasso Regression 3



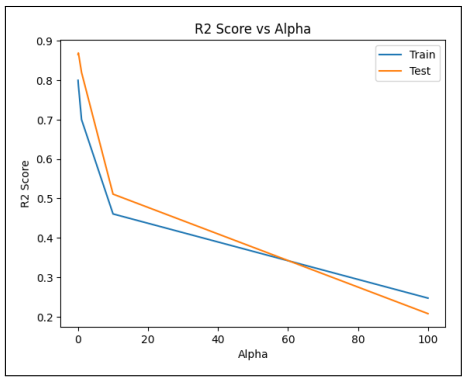
ElasticNet Regression

Hình 3.1: Tổng hợp kết quả dự đoán

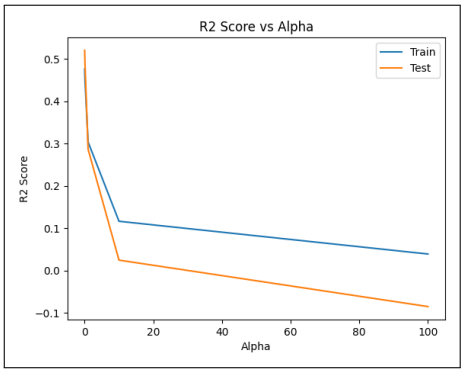
Có thể thấy, mỗi mô hình đưa ra đều là đường thẳng, nhưng các chúng xử lý các trọng số và đặc trưng sẽ khác nhau:

- Least Square Estimation: nó nằm gần như ở chính giữa cụm dữ liệu không cần biết là có phải dữ liệu nhiễu hay không
- Ridge Regression: đường thẳng dự đoán có thể ít nhạy cảm hơn với các đặc trưng ít quan trọng
- Lasso Regression: nó đã loại bỏ một số đặc trưng khỏi mô hình và dự đoán có thể ít nhạy cảm hơn với các đặc trưng ít quan trọng và có thể không bao gồm chúng
- ElasticNet Regression: dự đoán ít nhạy cảm hơn với các đặc trưng ít quan trọng và có thể không bao gồm một số đặc trưng như Lasso

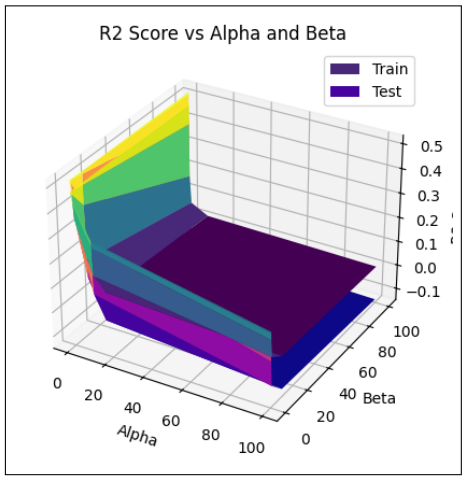
Khảo sát sự tương quan của tham số α, β của l1, l2 với Lasso, Ridge và cả l1, l2 với ElasticNet, ta có đồ thị:



L1 penality



L2 penality



L1 + L2

Hình 3.2: Khảo sát

Theo tính chất lý thuyết đã nêu bên trên, qua đồ thị dễ thấy rằng việc chọn tham số phù hợp chính là vùng đồ thị giữa train và test giao nhau.

Ma trận đánh giá

Qua bảng trên, với data này ElasticNet là phương pháp tối ưu nhất. Bằng thư viện của Sklearn, ta có thể tìm được tham số phù hợp cho mỗi phươgn pháp trên.

3.2.2 Tối ưu mô hình Logistic Regression

Mục này là mục mở rộng, người đọc có thể tìm đọc ở link git được gắn cuối bài báo cáo.

	0	1	2
0	Optimier	MSE	R2 Score
1	Ridge Regression	0.56376	0.43624
2	Lasso Regression	0.675051	0.324949
3	Elastic Net Regression	0.56056	0.43944

Hình 3.3: Chỉ số MSE, R2 score

```
Ridge Regression:
Best Parameters: {'alpha': 0.1}
Best Score: 0.5176925185046739

Lasso Regression:
Best Parameters: {'alpha': 0.01}
Best Score: 0.5149505455473985

ElasticNet Regression:
Best Parameters: {'alpha': 0.01, 'l1_ratio': 0.1}
Best Score: 0.5226549250649263
```

Hình 3.4: Tìm tham số tối ưu bằng GridSearch

Chương 4

Kết luận

4.1 Tự đánh giá

1. Mục tiêu:

    - Mục tiêu của báo cáo này là tìm hiểu các hướng phát triển cũng như so sánh các thuật toán hiện thời liên quan tới hồi quy tuyến tính trong học máy.

2. Cơ sở dữ liệu:

    - Bộ dữ liệu chúng tôi sử dụng được thu thập từ kaggle.com. Nội dung nói về các công thức nấu ăn và giá trị dinh dưỡng chứa trong chúng.
    - Bộ dữ liệu này hiện chứa 17736 dòng chứa các công thức nấu ăn, mỗi dòng chứa 669 biến là các giá trị 0 và 1 thể hiện sự xuất hiện hoặc không của từng nguyên liệu.

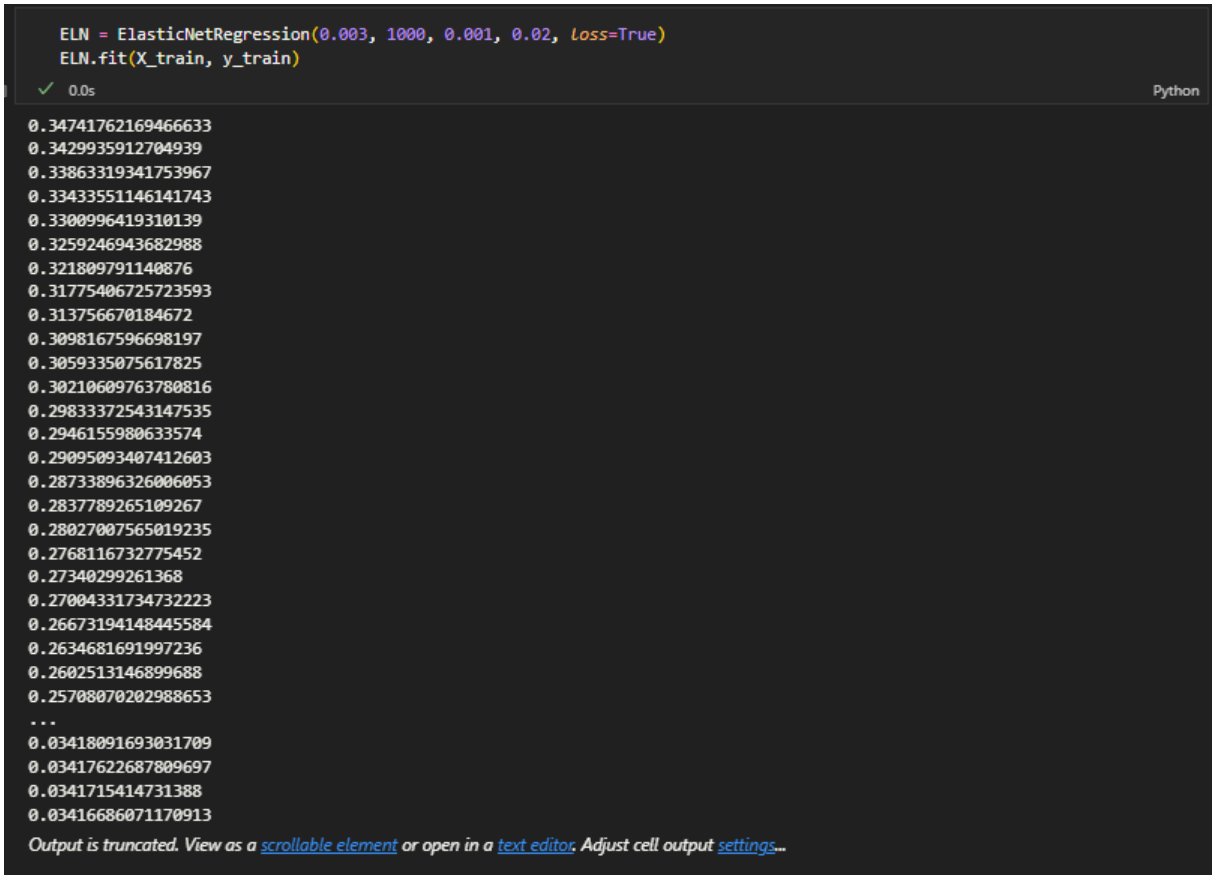
3. Ma trận đánh giá:

    - MSE
    - R2 score

4.2 Kết quả thực nghiệm

Mô hình của chúng tôi sẽ được đánh giá trên bộ dữ liệu bao gồm hơn 16000 điểm dữ liệu với gần 700 chiều. Chúng tôi thu thập dữ liệu từ Kaggle và đảm bảo dữ liệu phù hợp vs bài toán chúng tôi đặt ra. Trước khi đào tạo mô hình, chúng tôi quyết định loại bỏ những cột không cần thiết và những dòng chứa thông tin bị thiếu hoặc lỗi.

Để dễ hình dung, đây là kết quả chạy mô hình:



4.3 Hướng phát triển dự án trong tương lai

Việc tìm hướng tối ưu hoá các phương pháp giải bài toán phân loại có thể là một chủ đề phức tạp và đòi hỏi tính chuyên môn, vì nó có khả năng duy trì thành kiến và thiên vị. Dưới đây là một số cải tiến tiềm năng trong tương lai khi tôi ưu một bài toán phân loại:

1. **Giảm thiểu sai lệch**: Giảm thiểu thiên vị trong học máy là một quá trình quan trọng để đảm bảo công bằng và chính xác trong dự đoán. Thiên vị có thể xuất hiện ở các giai đoạn khác nhau của quá trình phát triển mô hình, bao gồm dữ liệu không đủ, thu thập dữ liệu không nhất quán và xử lý dữ liệu kém. Việc xác định, đánh giá và loại bỏ bất kỳ thiên vị nào có thể ảnh hưởng đến dự đoán là rất quan trọng. Tuy thiên vị không thể được giải quyết hoàn toàn nhưng nó có thể được giảm xuống mức tối thiểu để có sự cân đối giữa thiên vị và sai số.

2. **Tăng độ chính xác**: Một lĩnh vực khác cần cải thiện là độ chính xác. Nghiên cứu trong tương lai có thể tập trung vào việc phát triển các thuật toán phù hợp hơn để phân loại, hoặc xử lý dữ liệu tốt hơn và đa dạng hơn.

Chương 5

Tài liệu liên quan

5.1 Sách kham khảo

- https://www.math.uci.edu/ qnie/Publications/NumericalOptimization.pdf
- https://github.com/tiepvupsu/ebookMLCB
- https://www.mbit.edu.in/wp-content/uploads/2020/05/Numerical$_m ethods_f or_e ngineers_f or_e ngi.pdf$

5.2 Source code

- Link git: https://github.com/Minhdd1413/Numerical-methods-for-machine-learning/tree/master