SỞ GIÁO DỤC VÀ ĐÀO TẠO THÀNH PHỐ HỒ CHÍ MINH
THPT CHUYÊN TRẦN ĐẠI NGHĨA

BÁO CÁO:

LINEAR REGRESSION

Giáo viên hướng dẫn: Th.S Hồ Ngọc Lâm

Nhóm học sinh thực hiện:

Trần Nguyễn Thanh Phương

Cao Minh Khoa

Vũ Minh Triết

Phan Minh Quân

Học sinh lớp: 12CTin – Khóa 2021-2024

Thành phố Hồ Chí Minh, ngày 17 tháng 10 năm 2023

Mục lục

1. Giới thuyết chung về Linear Regression (Hồi Quy Tuyến Tính)

1.1. Thuật ngữ Hồi quy trong Hồi Quy Tuyến Tính

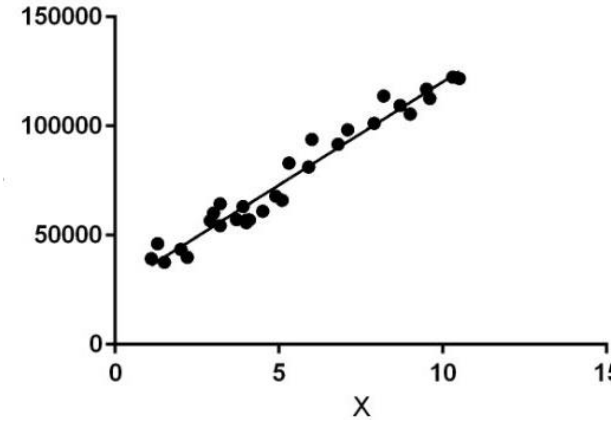
Để làm rõ về mô hình hồi quy tuyến tính, trước tiên ta cần tìm hiểu về khái niệm hồi quy.Hồi quy chính là một phương pháp thống kê để thiết lập một mối quan hệ giữa một biến phụ thuộc và một nhóm tập hợp các biến độc lập. Một ví dụ cơ bản của hồi quy đó chính là bài toán thiết lập một mối quan hệ giữa chiều cao và cân nặng của một người với tuổi của người đó, mô hình trên có thể biểu diễn dưới dạng phương trình như sau:

$$Tuổi = Chiều\ cao \times 10 + Cân\ nặng * 20 + 9$$

1.2. Khái niệm Linear Regression (Hồi Quy Tuyến Tính)

Linear Regression (Hồi Quy Tuyến Tính) là một trong những thuật toán cơ bản và phổ biến nhất trong Machine Learning (Học máy) nói chung và Supervised Learning (học có giám sát) nói riêng. Bởi quan hệ giữa đầu vào và đầu ra được mô tả thông qua một hàm tuyến tính nên đầu ra được dự đoán là liên tục. Thuật toán này thích hợp để dự đoán doanh số hay giá cả thay vì phân loại chúng thành các đại lượng rời rạc như hình dạng, kích thước, màu sắc hay xác định đối tượng như mèo, chó,...

Trong Linear Regression, có hai loại bài toán chính đó là hồi quy tuyến tính đơn biến và hồi quy tuyến tính đa biến. Univariate Linear Regression (Hồi Quy Tuyến Tính Đơn Biến) là mối quan hệ giữa hai biến số liên tục trên trục hoành x và y. Phương trình hồi quy tuyến tính đơn biến có dạng $y = ax + b$ với x là biến độc lập và y là biến phụ thuộc vào x. Với Multivariate Linear Regression (Hồi Quy Tuyến Tính Đa Biến), đây có thể hiểu là một mô hình hồi quy có nhiều biến độc lập $x_1, x_2, x_3, \ldots, x_N$ và nhiều hệ số $a_1, a_2, a_3, \ldots, a_N$ thay vì một biến độc lập x duy nhất. Mục tiêu của thuật toán này chính là tìm được một hàm tuyến tính trả về kết quả là một biến y phụ thuộc dựa trên các biến x độc lập. Khi biểu diễn hàm tuyến tính này trên không gian 2 chiều, ta nhận được một đường thẳng biểu diễn mối quan hệ giữa đầu vào và đầu ra. Độ dốc của đường thẳng này cho ta biết biến phụ thuộc y thay đổi bao nhiêu khi các biến độc lập x thay đổi.



Hình 1.1 Đồ thị minh họa của một đường thẳng Linear Regression trên hệ trục tọa độ Oxy

Đường thẳng trong đồ thị minh họa trên biểu thị sự quan hệ giữa biến độc lập x và biến phụ thuộc dự đoán y. Hàm Hồi Quy Tuyến Tính sẽ trả trả về một đường thẳng gần đúng nhất dựa trên các điểm trong tập dữ liệu được đưa vào.

1.3. Mô hình của bài toán Linear Regression (Hồi Quy Tuyến Tính)

Giả sử $X_1, X_2, \ldots, X_N$ là N biến độc lập dùng để dự báo. Y là biến phụ thuộc cần dự báo. Ta có mô hình hồi quy tuyến tính như sau:

$$Y \approx \hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_N X_N + \varepsilon$$

với ε là sai số ngẫu nhiên, $\beta_0, \beta_1, \beta_2, \ldots, \beta_N$ là các hệ số chưa biết. Ta tiến hành k lần quan sát $N + 1$ biến độc lập:

$$y_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_N x_{1N} + \varepsilon_1$$
$$y_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \ldots + \beta_N x_{2N} + \varepsilon_2$$
$$y_3 = \beta_0 + \beta_3 x_{31} + \beta_2 x_{32} + \ldots + \beta_N x_{3N} + \varepsilon_3$$
$$\ldots$$
$$y_k = \beta_0 + \beta_1 x_{k1} + \beta_2 x_{k2} + \ldots + \beta_N x_{Nk} + \varepsilon_k$$

Mô hình trên có thể viết lại như sau:

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1N} \\ 1 & x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{k1} & x_{k2} & x_{k3} & x_{kN} \end{bmatrix} + \begin{bmatrix} \beta_0 \\ \beta_1 \\ \ddots \\ \beta_N \end{bmatrix} + \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \ddots \\ \varepsilon_k \end{bmatrix}$$

2. Cơ sở lý thuyết

2.1. Sai số dự đoán

Mục tiêu mà chúng ta hướng tới khi xây dựng biểu thức sai số dự đoán $\boldsymbol{e}$ là để độ chênh lệch giữa giá trị thực $\boldsymbol{y}$ và giá trị dự đoán $\hat{\boldsymbol{y}}$ là nhỏ nhất. Tức là ta muốn giá trị biểu thức sau đây càng nhỏ càng tốt.

$$\frac{1}{2}\boldsymbol{e}^2 = \frac{1}{2}(\boldsymbol{y} - \hat{\boldsymbol{y}})^2 = \frac{1}{2}(\boldsymbol{y} - \bar{\boldsymbol{w}}\boldsymbol{x} + \bar{\boldsymbol{b}})^2$$

trong đó hệ số $\frac{1}{2}$ là để thuận tiện cho việc tính toán. Chúng ta cần $\boldsymbol{e}^2$ vì $\boldsymbol{e} = \boldsymbol{y} - \bar{\boldsymbol{y}}$ có thể là một số âm, việc nói $\boldsymbol{e}$ nhỏ nhất sẽ không đúng vì khi $\boldsymbol{e} = -\infty$ là rất nhỏ nhưng sự sai lệch là rất lớn. Để xử lý vấn đề trên ta đề xuất một phương án sử dụng hàm mất mát Loss Function.

2.2. Loss Function (Hàm mất mát)

2.2.1. Định nghĩa Loss Function

Loss Function kí hiệu là $\boldsymbol{L}$, là thành phần chính của hàm đánh giá và hàm mục tiêu. Cụ thể, trong công thức thường gặp ở hàm Loss Function $\boldsymbol{L}$:

$$\boldsymbol{L_D}(\boldsymbol{f_w}) = \frac{1}{|\boldsymbol{D}|}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{D}} \boldsymbol{L}(\boldsymbol{f_w}(\boldsymbol{x}), \boldsymbol{y})$$

Loss Function trả về một số thực không âm thể hiện sự chênh lệch giữa hai đại lượng: $\hat{\boldsymbol{y}}$, giá trị được dự đoán và $\boldsymbol{y}$, giá trị thực tế. Loss Function giống như một hình thức để bắt model đóng phạt mỗi lần nó dự đoán sai, và số mức phạt tỉ lệ thuận với độ trầm trọng của sai sót. Trong mọi bài toán supervised learning, mục tiêu của ta luôn bao gồm giảm thiểu tổng mức phạt phải đóng. Trong trường hợp lý tưởng $\hat{\boldsymbol{y}} = \boldsymbol{y}$, Loss Function sẽ trả về giá trị cực tiểu bằng 0.

Về mặt bản chất, Loss Function là một phương pháp đo lường chất lượng của mô hình dự đoán trên tập dữ liệu quan sát. Nếu mô hình dự đoán sai nhiều thì giá trị của Loss Function sẽ càng lớn và ngược lại nếu nó dự đoán càng đúng thì giá trị của Loss Function sẽ càng nhỏ.

2.2.2. Cách xây dựng Loss Function

Vì Loss Function đo đạc chênh lệch giữa $\boldsymbol{y}$ và $\hat{\boldsymbol{y}}$, nên không lạ gì nếu ta nghĩ ngay đến việc lấy hiệu giữa chúng:

$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \hat{\boldsymbol{y}} - \boldsymbol{y}$$

Tuy nhiên hàm này lại không thỏa mãn tính chất không âm của một Loss Function. Ta có thể sửa nó lại để thỏa mãn tính chất này. Ví dụ như lấy giá trị tuyệt đối của hiệu:

$$\boldsymbol{L(\hat{y}, y) = |\hat{y} - y|}$$

Loss Function này không âm nhưng lại không thuận tiện trong việc cực tiểu hóa, bởi vì đạo hàm của nó không liên tục (nhớ là đạo hàm của $|\hat{y} - y|$ bị đứt quãng tại $\hat{y} - y = 0$) và thường các phương pháp cực tiểu hóa hàm số thông dụng đòi hỏi phải tính được đạo hàm. Một cách khác đó là lấy bình phương của hiệu:

$$\boldsymbol{L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2}$$

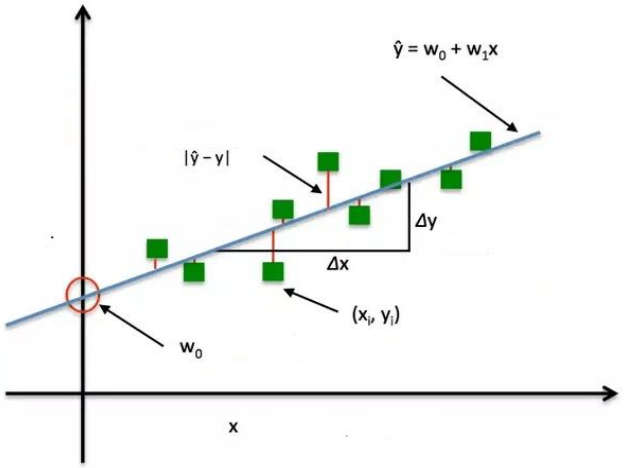
Khi tính đạo hàm theo $\hat{y}$, ta được $\boldsymbol{\partial L = \frac{1}{2} \times 2 \times (\hat{y} - y) = \hat{y} - y}$. Các bạn có thể thấy rằng hằng số $\frac{1}{2}$ được thêm vào chỉ để cho công thức đạo hàm được đẹp hơn, không có hằng số phụ. Loss Function này được gọi là Square Loss.

2.2.3. Ví dụ minh họa

Ví dụ 1: Ở đây, chúng ta định nghĩa một hàm Loss Function đơn giản như sau. Xét bài toán hồi quy mức độ hài lòng với công việc theo mức lương. Với mỗi dự đoán mà mô hình đưa ra, chúng ta đưa ra một giá trị mất mát do dự đoán sai là trị tuyệt đối của hiệu giữa giá trị dự đoán và giá trị thực trong bộ dữ liệu quan sát. Hay Loss $= |\hat{y} - y|$, trong đó $\hat{y}$ là giá trị dự đoán của mô hình và y là giá trị thật được lấy đúng trong bộ dữ liệu quan sát. Lấy một số điểm dự đoán ta tính được giá trị mất mát tại một số điểm trong bộ dữ liệu quan sát như sau:

Giá trị thật y	Giá trị dự báo $\hat{y}$	Giá trị mất mát do dự đoán sai
60	60	0
70	80	10
80	70	10

Như vậy, chúng ta có thể tính được giá trị mất mát do mô hình dự đoán tại tất cả các điểm trong bộ dữ liệu quan sát. Giá trị này có thể lớn, có thể nhỏ và khá rời rạc nên chúng ta không thể biết được đâu là dự đoán tốt, đâu là dự đoán không tốt, vì vậy để đánh giá mức độ tốt của mô hình dự đoán, chúng ta lấy trung bình giá trị mất mát của toàn bộ các điểm trên bộ dữ liệu quan sát. Điều này giúp ta xác định được một Loss Function trên toàn bộ bộ dữ liệu quan sát:



Hình 2.1 Đồ thị biểu diễn phương trình đường thẳng và mối tương quan giữa giá trị dự đoán và giá trị thực

$$L = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

2.2.4. Tiểu kết

Nói tóm lại, Loss Function là các hàm phạt, đo lường mức độ dự đoán sai của mô hình trên dữ liệu quan sát hữu hạn, nó là trung bình lỗi dự đoán trên toàn bộ bộ dữ liệu và thường cần nhận giá trị thực, không âm (để đo cường độ và không làm triệt tiêu lỗi khi tính trung bình), ngoài ra nó nên có đạo hàm dễ xác định để tiện cho việc tối ưu hóa sử dụng các thuật toán dựa trên đạo hàm hoặc gradient sau này.

Các Loss Function này có thể được sử dụng trong cả quá trình huấn luyện lẫn quá trình đánh giá mô hình. Trong quá trình đánh giá mô hình nó được gọi là hàm đánh giá (evaluation Function). Còn ngược lại khi chúng ta sử dụng các Loss Function này vào quá trình huấn luyện trên tập dữ liệu huấn luyện, chúng ta gọi nó là objective Function hay hàm mục tiêu mà chúng ta cần tối ưu hóa.

2.2.5. Các bước cơ bản để tìm nghiệm cho bài toán Linear Regression

- Chuẩn bị dữ liệu:

– Thu thập dữ liệu: Thu thập tất cả dữ liệu liên quan, bao gồm biến độc lập (đầu vào) và biến phụ thuộc (đầu ra).

– Khám phá dữ liệu (Exploratory Data Analysis - EDA): Thực hiện EDA để hiểu rõ cấu trúc và mối quan hệ trong dữ liệu, xem xét phân phối, kiểm tra giá trị bị thiếu và ngoại lệ.

- Tách dữ liệu:

- Chia dữ liệu thành hai phần: tập huấn luyện (training set) và tập kiểm tra (test set). Tập huấn luyện được sử dụng để xây dựng mô hình, trong khi tập kiểm tra được sử dụng để đánh giá hiệu suất của mô hình.

- Chọn mô hình Linear Regression:

  - Linear Regression có thể được biểu diễn dưới dạng phương trình đường thẳng: $y = wx + b$, trong đó y là biến phụ thuộc, x là biến độc lập, w là hệ số góc (slope), và b là hệ số giao điểm (intercept).

- Xây dựng mô hình:

  - Sử dụng tập huấn luyện để học các giá trị tối ưu cho w và b. Các giá trị này được tìm thông qua phương pháp hồi quy tối thiểu bình phương (Least Square Regression), trong đó mục tiêu là tối thiểu hóa tổng bình phương sai số giữa dự đoán và thực tế.

- Đánh giá mô hình:

  - Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình. Các phép đo phổ biến bao gồm Mean Squared Error (MSE), Root Mean Squared Error (RMSE), và R-squared (R^2).

- Tinh chỉnh mô hình (tuỳ chọn):

  - Nếu mô hình không đạt kết quả tốt, bạn có thể cân nhắc tối ưu hóa các siêu tham số, thêm biến độc lập mới, loại bỏ biến không quan trọng, hoặc sử dụng các biến đổi dữ liệu.

- Sử dụng mô hình:

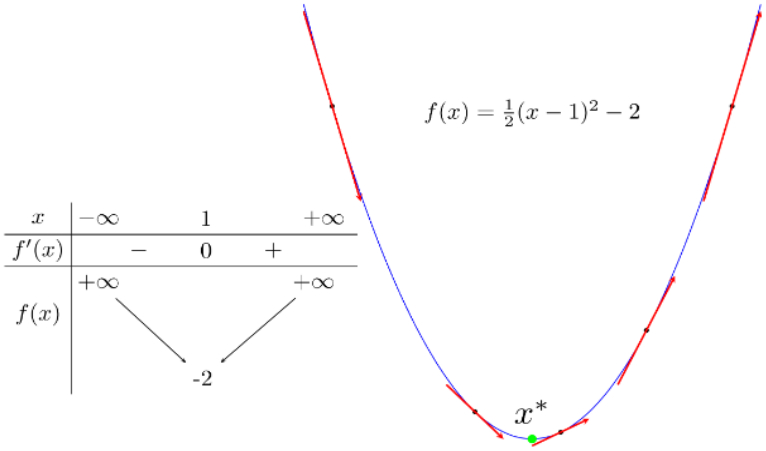
  - Khi bạn hài lòng với hiệu suất của mô hình, bạn có thể sử dụng nó để thực hiện dự đoán cho giá trị đầu ra mới dựa trên giá trị đầu vào.

- Giám sát mô hình:

  - Theo dõi mô hình sau khi triển khai để đảm bảo rằng nó vẫn phù hợp với dữ liệu mới và hiệu suất của nó không giảm đi theo thời gian.

  - Nếu cần tối ưu hóa mô hình hoặc xử lý các tình huống đặc biệt, bạn có thể sử dụng các biến thể của Linear Regression như Ridge Regression, Lasso Regression hoặc Elastic Net.

2.3. Gradient Descent trong Linear Regression



Hình 2.2 Khảo sát sự biến thiên của một hàm số bậc hai

Gradient descent (suy giảm độ dốc) là một thuật toán tối ưu hóa lặp để tìm cực trị của một hàm. Để tìm cực tiểu cục bộ (đạo hàm bằng không, đạo hàm bên phải không dương và đạo hàm bên trái không âm) của một hàm, người ta có thể thực thiện các bước tỷ lệ thuận với âm của gradient (độ dốc xấp xỉ) của hàm hiện tại.

Như ta có thể thấy, hàm $f(x) = \frac{1}{2}(x-1)^2 - 2$ là một hàm khả vi có cực trị tại 2, việc tìm cực trị của hàm này là rất đơn giản và không cần những bài toán phức tạp (ta chỉ cần tìm nghiệm $f'(x) = 0$, tuy nhiên trong thực tế, giải bài toán này một cách trực tiếp gần như là bất khả thi do sự phức tạp của những bài toán đạo hàm hoặc do có quá nhiều chiều dữ liệu.

Gradient descent là quá trình xuất phát từ một điểm mà chúng ta coi là gần với nghiệm của bài toán, sau đó dùng phép toán lặp để tiến lại gần điểm cần tìm (đạo hàm bằng không). Chúng ta sẽ tìm đạo hàm tại một điểm khởi đầu ngẫu nhiên (hoặc cài sẵn), và đi ngược dấu đạo hàm. Với ví dụ trên, nếu $x < 1$, đạo hàm dấu âm và ta cần phải di chuyển giá trị về chiều x dương, nếu $x > 1$, đạo hàm dấu dương và ta cần phải di chuyển giá trị về chiều x âm cho đến khi đạo hàm bằng không tại $x = 1$.

Trong Linear Regression, Gradient Descent được dùng để tối ưu hàm mất mát :

$$L(w,b)_i = \frac{1}{2n}\sum_{k=1}^{n}(y_k - (w_i x_k + b_i))^2$$

bằng cách lặp lại quá trình tổng quát sau:

$$w_{i+1} = w_i - \alpha\frac{\partial L}{\partial w} \text{ và } b_{i+1} = b_i - \alpha\frac{\partial L}{\partial b}$$

với phương trình đường thẳng là $y_i = w_i x + b_i$ và đạo hàm của Loss Function theo hai biến lần lượt là:

- Đạo hàm của Loss Function theo w

$$\frac{\partial L}{\partial w}(\frac{1}{2n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))^2)$$

- Đạo hàm của Loss Function theo b

$$\frac{\partial L}{\partial b}(\frac{1}{2n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))^2)$$

với α là tốc độ học.

Trong đây, $\frac{\partial L}{\partial w}$ và $\frac{\partial L}{\partial b}$ là đạo hàm của phương trình này, và việc thay đổi giá trị cần tìm bằng cách đi ngược những đạo hàm này chính là sự suy giảm độ dốc của một đồ thị hàm số.

2.3.1. Tìm nghiệm của bài toán Linear Regression qua Gradient Descent

– Đạo hàm của Loss Function theo w:

$$\frac{\partial L}{\partial w}\left(\frac{1}{2n}\sum_{k=1}^{n}\left(y_k-(w_i x_k+b_i)\right)^2\right)$$

$$=\frac{1}{n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))\frac{\partial L}{\partial w}(y_k-(w_i x_k+b_i))$$

$$=\frac{1}{n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))\,(0-x_k+0)$$

$$=\frac{1}{n}\sum_{k=1}^{n}(w_i x_k+b_i-y_k)\,(x_k)$$

- Đạo hàm của Loss Function theo b:

$$\frac{\partial L}{\partial b}\left(\frac{1}{2n}\sum_{k=1}^{n}\left(y_k-(w_i x_k+b_i)\right)^2\right)$$

$$=\frac{1}{n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))\frac{\partial L}{\partial b}(y_k-(w_i x_k+b_i))$$

$$=\frac{1}{n}\sum_{k=1}^{n}(y_k-(w_i x_k+b_i))\,(0-0-1)$$

$$=\frac{1}{n}\sum_{k=1}^{n}(w_i x_k+b_i-y_k)$$

3. Minh họa giải thuật

3.1. Bước thực hiện

Để tìm được đường hồi quy tốt nhất, cần phải tìm được các tham số cho phương trình $y_i = w_i x + b_i$ sao cho giá trị hàm mất mát là thấp nhất có thể. Để làm việc này, ta cần tính đạo hàm của hàm mất mát sử dụng Gradient Descent, sau đó cập nhật tham số cho hàm số đường với α là tốc độ học của giải thuật

$$w_{i+1} = w_i - \alpha \frac{\partial L}{\partial w}$$

$$b_{i+1} = b_i - \alpha \frac{\partial C}{\partial b}$$

3.2. Cài đặt

3.2.1. Các bước

- **Huấn luyện (Training)**
  - Khởi tạo w và b với giá trị 0.
- **Xử lý (Process)**
  - Lấy dữ liệu tại x.
  - Dự đoán kết quả sử dụng $\hat{y} = wx + b$.
  - Tính độ mất mát.
  - Áp dụng Gradient descent để tính các tham số mới.
  - Lặp lại với tất cả giá trị trong bảng dữ liệu.
- **Áp dụng**
  - Sử dụng các công cụ lập trình thể hiện đường thẳng trên đồ thị.

3.2.2. Tối Ưu

Ta có thể tối ưu phương trình nhằm mục đích cài đặt :

- Đạo hàm của Loss Function theo w

$$\frac{\partial L}{\partial w} = \frac{1}{n}\sum_{k=1}^{n}\left((w_i x_k + b_i) - y_k\right)(x_k)$$

$$= \frac{\partial L}{\partial w} = \frac{1}{n}\sum_{k=1}^{n}(\hat{y} - y_k)(x_k)$$

- Đạo hàm của Loss Function theo b

$$\frac{\partial L}{\partial b} = \frac{1}{n}\sum_{k=1}^{n}(w_i x_k + b_i - y_k)$$

$$= \frac{\partial L}{\partial b} = \frac{1}{n}\sum_{k=1}^{n}(\hat{y} - y_k)$$

Nhằm tránh tính $\hat{y}$ cho tất cả giá trị x, ta có thể gọi X là ma trận chứa các giá trị, khi này, $\hat{y}$ có thể được tính như sau:

$$\hat{y} = wX + b \; với \; X = [x_1 \quad x_2 \quad \cdots \quad x_n]$$

$$wX = [wx_1 \quad wx_2 \quad \cdots \quad wx_n]$$

$$\hat{y} = [wx_1 + b \quad wx_2 + b \quad \cdots \quad wx_n + b]$$

Kết hợp hai phương thức trên, ta có thể tính được đạo hàm theo trọng số của hàm mất mát bằng cách nhân ma trận như sau:

$$\frac{\partial L}{\partial w} = X^T \cdot \hat{y} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \cdot [wx_1 + b \quad wx_2 + b \quad \cdots \quad wx_n + b]$$

3.2.3. Chi tiết cài đặt

3.2.3.1. Khởi tạo

Khởi tạo giải thuật với tốc độ dọc, giá trị phần tử khai báo weight và bias:

```
def __init__(self, lr = 0.001, n_iters=1000):
    self.lr = lr
    self.n_iters = n_iters
    self.weights = None
    self.bias = None
```

3.2.3.2. Xử lý

Ta tính được $\hat{y}$ trên toàn bộ dữ liệu:

```
y_pred = np.dot(X, self.weights) + self.bias
```

Sau đó, qua cải thiện công thức, ta tính được đạo hàm của hàm mất mát theo tham số trên tất cả giá trị bằng nhân ma trận. Tương tự, ta có thể tính đạo hàm của hàm số mất mát theo tham số b bằng công thức đã chứng minh tại **2.3.1**:

```
dw = (1/n_samples) * np.dot(X.T, (y_pred-y))
db = (1/n_samples) * np.sum(y_pred-y)
```

Khi này, ta có thể cập nhật các giá trị tham số theo các đạo hàm mới và tốc độ học:

```
self.weights = self.weights - self.lr * dw
self.bias = self.bias - self.lr * db
```

Lặp lại quá trình cho tất cả giá trị x trong dữ liệu có được đường thẳng cần tìm.

3.2.3.3. Áp dụng

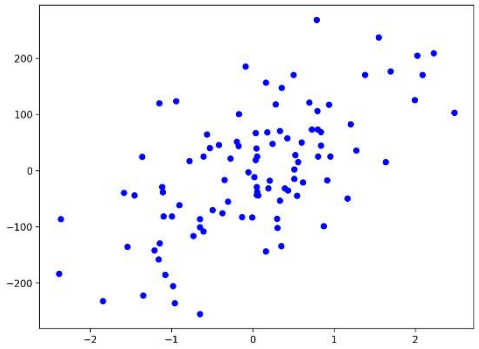
Sử dụng chức năng Dataset của thư viện SciKit Learn, ta sinh được 1 Dataset phù hợp cho việc trình bày hoạt động của "Hồi quy tuyến tính" với các điều kiện ràng buộc rõ ràng.

```
X, y = datasets.make_regression(n_samples=100, n_features=1, noise=100, random_state=4)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1234)
```

Cùng lúc đó, qua thư viện matplotlub.pyplot, ta sẽ trình bày dataset trên lên một hệ trục Oxy.

```
fig = plt.figure(figsize=(8,6))
plt.scatter(X[:, 0], y, color = "b", marker = "o", s = 30)
plt.show()
```

Hình ảnh minh họa của dataset trong một hệ trục Oxy:

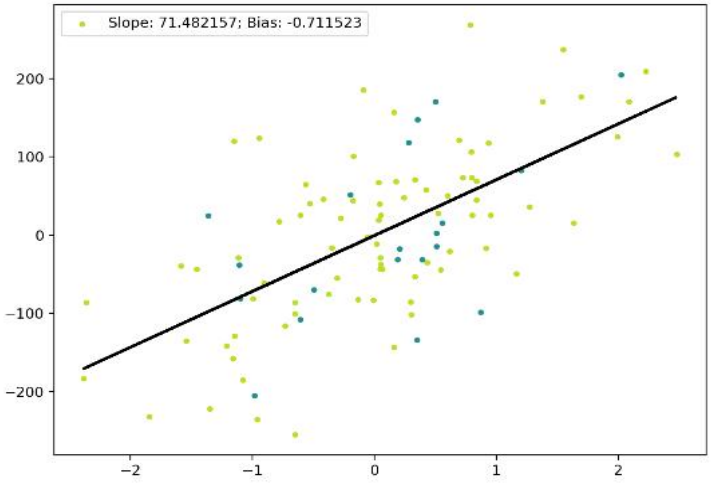


Hình 3.1 Hình ảnh minh họa của dataset trong hệ trục Oxy

Xác định được Dataset cần phân tích, ta thực hiện huấn luyện Hồi quy tuyến tính và nhận lại giá trị thông tin đường thẳng, tham w và tham số b của đường thẳng.

```
reg = LinearRegression(lr=0.01)
reg.fit(X_train,y_train)
y_pred_line,b,c = reg.predict(X)
```

Khi đó ta có thể vẽ được đường thẳng $\hat{y}$ trên hệ trục Oxy trước đó, để thể hiện được tính liên quan giữa các giá trị của Dataset và đường thẳng tìm được:



Hình 3.2 Kết quả thu được dựa trên dataset

4. Tài liệu tham khảo

[1] https://viblo.asia/p/linear-regression-hoi-quy-tuyen-tinh-trong-machine-learning-4P856akRlY3?fbclid=IwAR3z4Bqpv5PDb7AeynCyQm4JpdcTtGgdhvAUiDK25kKywsvMTdyIoUH-iXw*, truy cập lần cuối 11h04 ngày 16/10/2023.*

[2] http://tutorials.aiclub.cs.uit.edu.vn/index.php/2021/04/24/linear-regression/?fbclid=IwAR1FcJNlvZgx-1E0R5M-9JFydEENDbG0-1CudaBKfnOKNIFY3Dj_bdJiIfA*, truy cập lần cuối 0h14 ngày 17/10/2023.*

[3] https://www.geeksforgeeks.org/ml-linear-regression/?ref=lbp&fbclid=IwAR0tbG_LGPLrAH3v0DILazejnBGflFcxCmQt7qKACecprXn4kYgfzRn2UTU*, truy cập lần cuối 0h30 ngày 17/10/2023.*

[4] https://www.vietlod.com/hoi-quy-tuyen-tinh-stata/2*, truy cập lần cuối 1h27 ngày 17/10/2023.*

[5] https://machinelearningcoban.com/2016/12/28/linearregression/*, truy cập lần cuối 22h32 ngày 17/10/2023.*

[6] Nguồn hình 1.1: https://www.geeksforgeeks.org/ml-linear-regression/?ref=lbp&fbclid=IwAR0tbG_LGPLrAH3v0DILazejnBGflFcxCmQt7qKACecprXn4kYgfzRn2UTU*, truy cập lần cuối 20h22 ngày 17/10/2023.*

[7] Nguồn hình 2.4: https://machinelearningcoban.com/2017/01/16/gradientdescent2/*, truy cập lần cuối 20h30 ngày 17/10/2023.*