**Công tác thí nghiệm số 1. Phần 3**

**Phần 3. Hồi quy Lasso và Ridge .**

Mục tiêu: củng cố kiến thức lý thuyết về hồi quy tuyến tính, rèn luyện kỹ năng thực hành xây dựng mô hình hồi quy tuyến tính.

Nhiệm vụ:

* Ghi nhớ và hiểu các khái niệm cơ bản của mô hình hồi quy tuyến tính;
* Tìm hiểu cách xây dựng mô hình hồi quy tuyến tính với chính quy L 1 và L 2;
* Tìm hiểu cách sử dụng thư viện scikit - tìm hiểu ;

**Hồi quy Lasso**

Lasso hồi quy ( Ít nhất tuyệt đối co ngót Và Lựa chọn Toán tử là một phương pháp hồi quy tuyến tính sử dụng chính quy hóa để giảm việc trang bị quá mức và lựa chọn tính năng. Nó thêm một hình phạt đối với các giá trị tuyệt đối của các hệ số vào hàm mất mát, khiến một số hệ số trở thành 0. Điều này làm cho Lasso trở nên hữu ích cho việc lựa chọn tính năng, đặc biệt khi bạn có nhiều biến số.

Lasso sử dụng chính quy L1, cộng tổng các giá trị tuyệt đối của các hệ số vào hàm mất:

trong đó RSS là tổng của số dư bình phương, là tham số chính quy hóa và β j​ là các hệ số mô hình.

Vì Lasso có thể đặt một số hệ số về 0 nên nó tự động chọn một tập hợp con các tính năng, giúp nó hữu ích khi làm việc với dữ liệu nhiều chiều . Hồi quy Lasso đối phó tốt với các tình huống trong đó các đặc điểm có tương quan với nhau ( đa cộng tuyến ), điều này có thể dẫn đến ước tính hệ số không ổn định trong hồi quy tuyến tính thông thường.

Tham số λ kiểm soát mức độ chính quy hóa. Giá trị lớn hơn dẫn đến chính quy hóa mạnh hơn và nhiều hệ số null hơn.

mô hình Hồi quy Lasso bằng thư viện NumPy .

Đầu tiên, hãy nhập các thư viện:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Trước tiên, hãy tạo dữ liệu tổng hợp, như chúng ta đã làm trong các phần trước của công việc thí nghiệm. Tuy nhiên, hãy thêm mối quan hệ tuyến tính giữa hai yếu tố độc lập, vì việc chuẩn hóa L 1 giúp cải thiện kết quả của mô hình kết quả bằng cách đưa về 0

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Tiếp theo, chúng ta sẽ sử dụng phương pháp giảm độ dốc để thu được hệ số hồi quy:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Hàm lasso\_gradient\_descent , thực hiện hồi quy Lasso bằng cách sử dụng phương pháp giảm độ dốc. Hàm này cập nhật các hệ số mô hình có tính đến việc chuẩn hóa L1.

**Thông số chức năng**

* **X** : Ma trận đặc điểm (thứ nguyên n×p , trong đó n là số lượng mẫu và p là số lượng đặc điểm).
* **y** : Vector của biến mục tiêu (thứ nguyên n ).
* **alpha** : Tham số chính quy kiểm soát mức độ chính quy L1.
* **learning\_rate** : Tốc độ học, xác định mức độ chúng tôi cập nhật các hệ số ở mỗi lần lặp.
* **n\_iterations** : Số lần lặp giảm dần độ dốc.

Đầu tiên, chúng tôi xác định số lượng mẫu ( n\_samples) và tính năng (n\_features ) , đồng thời khởi tạo vectơ hệ số thành số 0 .

Tiếp theo là một vòng lặp sẽ thực hiện một số lần lặp nhất định (các bước giảm dần độ dốc).

**Tính toán độ dốc**

phần dư = y - X @ hệ số

gradient = - ( 2 ) \* (XT @ phần dư)\* learning\_rate+alpha \* np.sign (hệ số)

* **R dư** . Chúng tôi tính toán phần dư (chênh lệch giữa giá trị thực và giá trị dự đoán). Điều này cho thấy các hệ số hiện tại mô tả dữ liệu tốt như thế nào.
* **Rực rỡ** . Chúng tôi tính toán độ dốc của hàm mất. Độ dốc hiển thị hướng mà các hệ số cần được thay đổi để giảm sai số. Chúng tôi lấy giá trị âm vì độ dốc biểu thị hướng mà hàm mất mát đang tăng và chúng tôi cần di chuyển theo hướng ngược lại.

Tại sao độ dốc được tính theo cách này? Hãy rút ra công thức gradient.

đâu là giá trị thực và là kết quả của phép tính sử dụng mô hình. , trong đó X là ma trận các đặc trưng (tham số của phương trình) và là các hệ số của phương trình.

Sau đó chúng ta viết lại hàm mất ở dạng ma trận:

Theo định nghĩa, gradient sẽ bằng:

Tiếp theo, chúng ta cập nhật các hệ số bằng cách trừ đi một giá trị tỷ lệ với gradient và tốc độ học. Đây là bước giảm độ dốc.

hệ số -= learning\_rate \* gradient

Chúng tôi thực hiện giáo dục :

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Hãy đánh giá kết quả:

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Như bạn có thể thấy, mô hình đã làm rất tốt việc xấp xỉ dữ liệu của chúng tôi. Ngoài ra, một trong những yếu tố độc lập được đặt bằng 0, loại bỏ hiện tượng đa cộng tuyến . Hãy so sánh kết quả nếu không có sự chính quy hóa ( hãy đặt alpha bằng 0).

Chúng tôi nhận được:

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Như bạn có thể thấy, mô hình này cũng xử lý được dữ liệu. Tuy nhiên, hiện tượng đa cộng tuyến vẫn tồn tại trong dữ liệu , dẫn đến kết quả ước tính không hiệu quả.

**Nhiệm vụ 1.**

1. Tải dữ liệu từ dữ liệu . csv . Điều này có thể được thực hiện bằng cách sử dụng NumPy hoặc Pandas , như chúng tôi đã thực hiện trong các tác phẩm trước.
2. Xây dựng mô hình cho dữ liệu này bằng mã hồi quy Lasso trước đó .

**Hồi quy Lasso scikit - tìm hiểu**

Lasso , lớp Lasso đã được phát triển trong thư viện Scikit - learn được phát triển cho python . Để sử dụng nó, bạn cần nhập nó như thế này:

từ sklearn.Tuyến\_model nhập Lasso

Tiếp theo, để xây dựng hàm hồi quy, bạn cần gọi hàm tạo của lớp này với tham số alpha bắt buộc . Tham số này là tham số λ từ công thức trên. Điều này sẽ tạo ra một đối tượng mô hình Lasso mà chúng ta phải lưu vào một biến:

lasso = Lasso(alpha=0.1) # alpha - tham số chính quy

Tiếp theo, chúng ta sử dụng phương pháp fit để huấn luyện mô hình:

lasso.fit (X, y)

Để có được các tham số tính toán của mô hình, bạn phải sử dụng cú pháp sau:

hệ số = lasso.coef\_

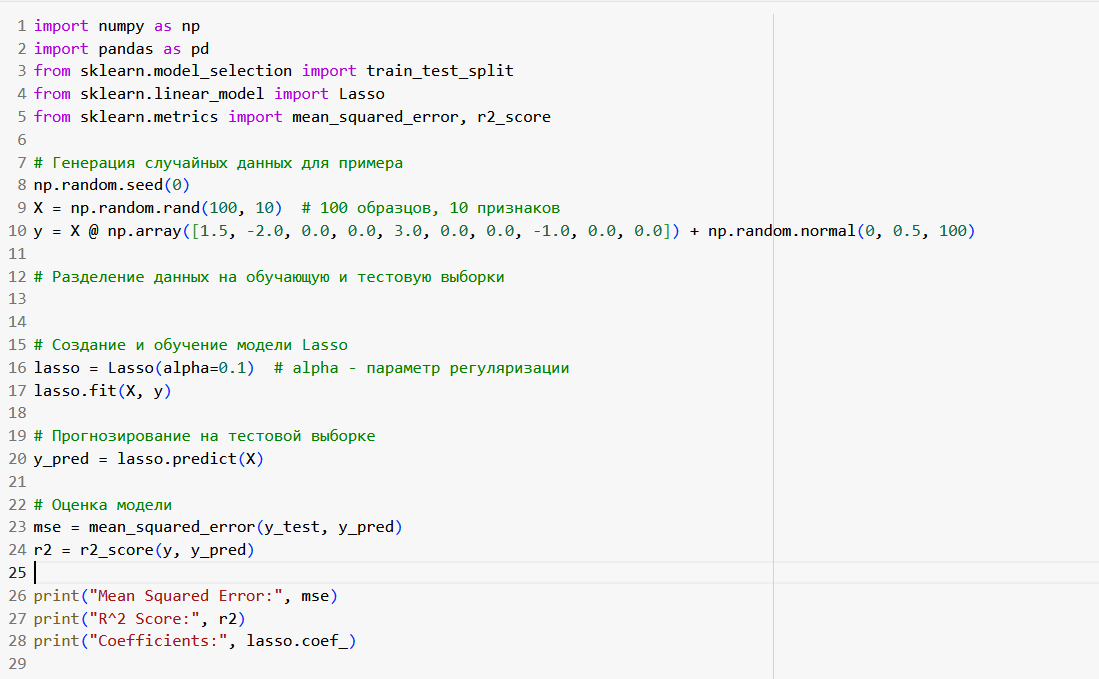
# hệ số đầu ra

print("Hệ số mô hình Lasso:", hệ số)

Để dự đoán các giá trị bằng mô hình đã xây dựng, phương pháp dự đoán được sử dụng :

y\_pred = lasso.predict ( X\_test )

Hãy xem một ví dụ mã:



1. **Nhập thư viện** . Chúng tôi nhập các thư viện cần thiết như numpy , pandas , train\_test\_split , Lasso và số liệu để đánh giá mô hình.
2. **Tạo dữ liệu** . Hãy tạo dữ liệu ngẫu nhiên chẳng hạn. Trong các vấn đề thực tế, bạn sẽ sử dụng bộ dữ liệu của riêng mình.
3. **Xây dựng và đào tạo mô hình** . Chúng tôi tạo một đối tượng Lasso với tham số alpha chính quy hóa nhất định và huấn luyện mô hình trên dữ liệu huấn luyện.
4. **Dự báo** . Chúng tôi sử dụng mô hình được đào tạo để dự đoán các giá trị trên mẫu thử nghiệm.
5. **Đánh giá mô hình** . Chúng tôi tính toán sai số bình phương trung bình (MSE) và hệ số xác định (R²) để đánh giá chất lượng của mô hình.
6. **Đầu ra của các hệ số** . Chúng tôi hiển thị các hệ số của mô hình để xem biến nào đã được chọn.

Bài tập

Theo ví dụ trên, tạo mô hình hồi quy Lasso cho dữ liệu boston . csv .

**Hồi quy sườn**

sườn là một phương pháp hồi quy tuyến tính bao gồm chính quy hóa L2. Nó được sử dụng để giải quyết vấn đề trang bị quá mức, đặc biệt khi bạn có nhiều tính năng (biến) hoặc khi các tính năng có tương quan với nhau. Ý tưởng cơ bản của hồi quy Ridge là thêm một hình phạt vào hàm mất mát phụ thuộc vào bình phương các hệ số của mô hình.

sườn thêm vào hàm mất mát (thường là sai số bình phương trung bình gốc) tổng bình phương của các hệ số nhân với một số tham số chính quy hóa (thường được ký hiệu là α). Nó trông như thế này:

Ridge không loại bỏ các hệ số như hồi quy Lasso (sử dụng chính quy L1). Thay vào đó, nó làm giảm giá trị của chúng, điều này có thể hữu ích khi tất cả các thuộc tính đều có giá trị nhưng bạn muốn giảm ảnh hưởng của chúng.

sườn đối phó tốt với các tình huống trong đó các đặc điểm có tương quan với nhau ( đa cộng tuyến ), điều này có thể dẫn đến ước tính hệ số không ổn định trong hồi quy tuyến tính thông thường.

Tham số α kiểm soát mức độ chính quy. Giá trị α cao hơn dẫn đến mức phạt lớn hơn đối với các hệ số lớn, điều này có thể giúp giảm tình trạng trang bị quá mức nhưng cũng có thể dẫn đến việc trang bị thiếu nếu giá trị quá lớn. Thông thường, α được chọn bằng phương pháp xác thực chéo.

Phương pháp bình phương tối thiểu trong trường hợp này sẽ được sửa đổi như sau:

β là vectơ của các hệ số,

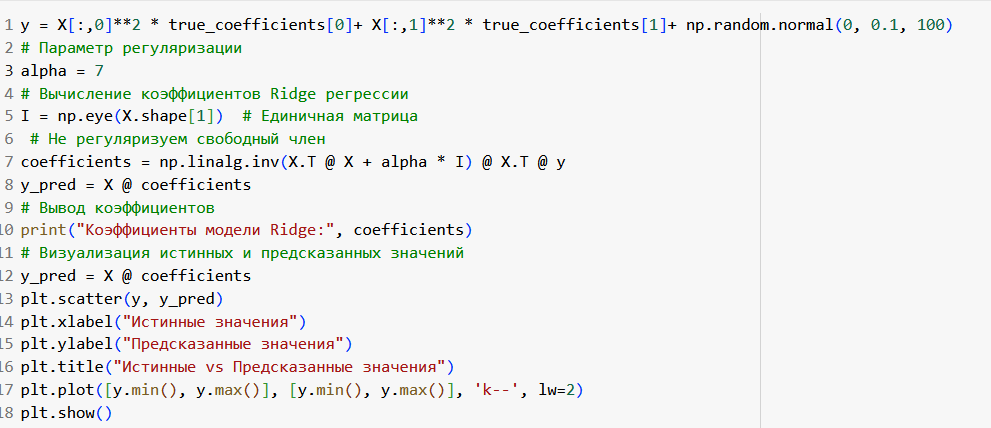
X - ma trận tính năng,

y là một vectơ của các giá trị đích,

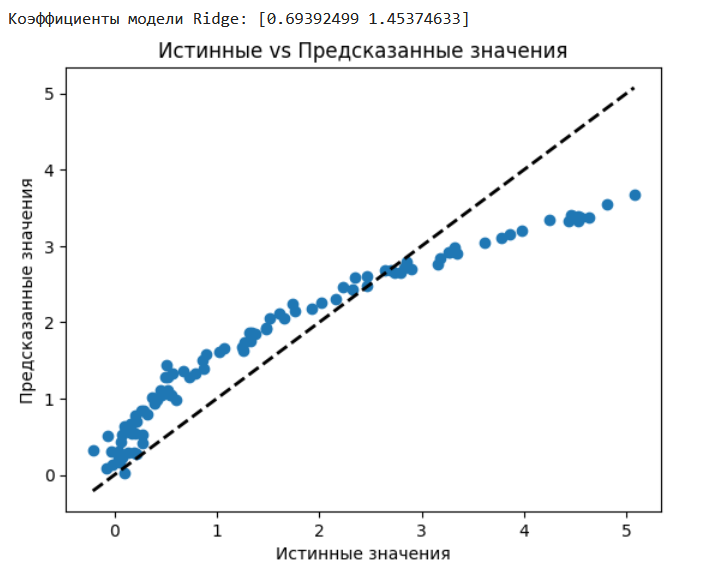
α là tham số chính quy,

Tôi là ma trận nhận dạng.

Triển khai trong python bằng cách sử dụng các biến tương tự, nhưng chúng tôi chọn dạng phụ thuộc đa thức (đa thức bậc hai). Khi đó code sẽ như sau:



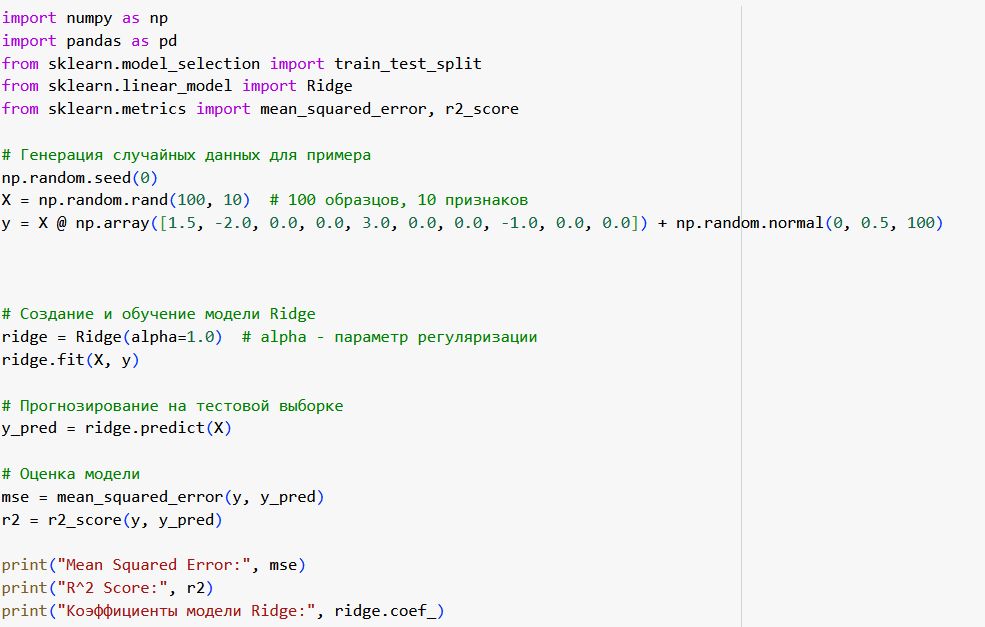
Kết quả là chúng tôi nhận được:



**sườn hồi quy scikit - tìm hiểu**

bạn có thể sử dụng thư viện scikit-learn để thực hiện hồi quy Ridge

Mã mẫu:



Bài tập

Theo ví dụ trên, tạo mô hình hồi quy Ridge cho dữ liệu boston . csv .

Nhiệm vụ cuối cùng

Xây dựng hồi quy Ridge và hồi quy Lasso cho bất động sản.

Câu hỏi bảo mật:

1. Ridge là gì ?
2. Lasso là gì ?
3. Chúng được sử dụng trong trường hợp nào?