**Лабораторная работа № 1. Ч.3**

**Часть 3. Регрессия Lasso и Ridge-регрессия.**

Цель: закрепить теоретические сведения о линейной регрессии, получить практические навыки построения модели линейной регрессии.

Задачи:

* Вспомнить и усвоить основные понятия моделей линейной регрессии;
* Научиться строить модели линейной регрессии с L1 и L2 регуляризации;
* Научиться использовать библиотеку scikit-learn;

**Регрессия Lasso**

Регрессия Lasso (Least Absolute Shrinkage and Selection Operator) — это метод линейной регрессии, который использует регуляризацию для уменьшения переобучения и выбора признаков. Она добавляет штраф за абсолютные значения коэффициентов к функции потерь, что приводит к тому, что некоторые коэффициенты могут стать равными нулю. Это делает Lasso полезным для выбора признаков, особенно когда у вас много переменных.

Lasso использует L1-регуляризацию, которая добавляет к функции потерь сумму абсолютных значений коэффициентов:

где RSS — это сумма квадратов остатков, — параметр регуляризации, а βj​ — коэффициенты модели.

Поскольку Lasso может устанавливать некоторые коэффициенты в ноль, он автоматически выбирает подмножество признаков, что делает его полезным для работы с высокоразмерными данными. Lasso регрессия хорошо справляется с ситуациями, когда признаки коррелируют друг с другом (мультиколлинеарность), что может привести к нестабильным оценкам коэффициентов в обычной линейной регрессии.

Параметр λ контролирует степень регуляризации. Большие значения приводят к более сильной регуляризации и большему количеству нулевых коэффициентов.

Рассмотрим практический пример построения модели Регрессии Lasso с использованием библиотеки NumPy.

Для начала импортируем библиотеки:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Для начала сгенерируем синтетические данные, как мы это делали в предыдущих частях лабораторной работы. Однако, добавим линейную зависимость между двумя независимыми факторами, так как L1 регуляризация помогает улучшать результаты полученной модели путем обнуления

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Далее воспользуемся методом градиентного спуска для получения коэффициентов регрессии:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Функция lasso\_gradient\_descent, которая реализует Lasso-регрессию с использованием градиентного спуска. Эта функция обновляет коэффициенты модели, учитывая L1-регуляризацию.

**Параметры функции**

* **X**: Матрица признаков (размерность n×p, где n — количество образцов, а p — количество признаков).
* **y**: Вектор целевой переменной (размерность n).
* **alpha**: Параметр регуляризации, который контролирует степень L1-регуляризации.
* **learning\_rate**: Скорость обучения, которая определяет, насколько сильно мы обновляем коэффициенты на каждой итерации.
* **n\_iterations**: Количество итераций градиентного спуска.

В начале мы определяем количество образцов (n\_samples) и признаков (n\_features), а также инициализируем вектор коэффициентов нулями.

Далее следует цикл, который будет выполняться заданное количество итераций (шаги градиентного спуска).

**Вычисление градиента**

residuals = y - X @ coefficients

        gradient = - (2) \* (X.T @ residuals)\*learning\_rate+alpha\*np.sign(coefficients)

* **Residuals**. Вычисляем остатки (разность между истинными значениями и предсказанными). Это показывает, насколько хорошо текущие коэффициенты описывают данные.
* **Gradient**. Вычисляем градиент функции потерь. Градиент показывает направление, в котором нужно изменять коэффициенты, чтобы уменьшить ошибку. Мы берем отрицательное значение, потому что градиент указывает направление увеличения функции потерь, а нам нужно двигаться в противоположном направлении.

Почему градиент вычисляется именно так? Выведем формулу градиента.

где — истинные значения, — результат вычисления с помощью модели. , где X – матрица признаков (параметры уравнения), – коэффициенты уравнения.

Тогда перепишем функцию потерь в матричном виде:

По определению градиент будет равен:

Далее обновляем коэффициенты вычитая из них значение, пропорциональное градиенту и скорости обучения. Это шаг градиентного спуска.

coefficients -= learning\_rate \* gradient

Выполняем обучение:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Оценим результат:

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Как видно, модель неплохо справилась с аппроксимацией наших данных. К тому же один из независимых факторов был обнулен, что исключило мультиколлинеарность. Давайте сравним с результатом, если бы не было регуляризации (alpha положим равным 0).

Получим:

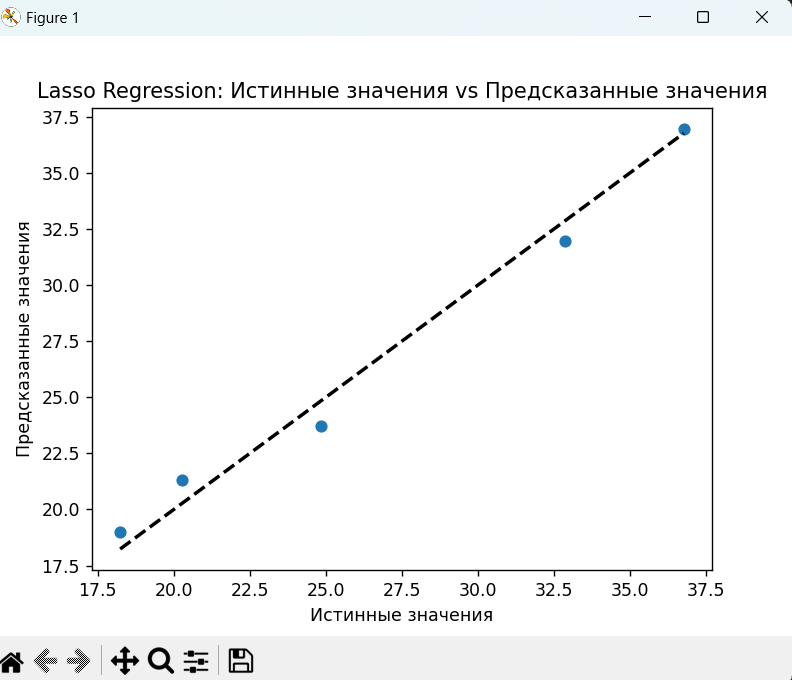
Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Как видно, данная модель тоже справилась с данными. Однако в данных осталась мультиколлинеарность, что приводит к неэффективности полученных оценок.

**Задание 1.**

1. Загрузите данные из data.csv. Это можно сделать либо с использованием NumPy, либо Pandas, как мы делали в предыдущих работах.
2. Постройте модель для этих данных, используя предыдущий код Lasso регрессию.



**Анализ результатов**

1. **Коэффициенты Lasso:** [2.96029858, 1.83558552]
   * Это означает, что оба признака (features) данных имеют важное значение, и модель Lasso сохранила их, не обнулив.
   * Эти коэффициенты были оптимизированы для минимизации функции ошибки с учетом L1-регуляризации.
2. **Среднеквадратичная ошибка (MSE):** 0.7459376275801016
   * Низкое значение MSE указывает на то, что разница между истинными значениями и предсказанными очень мала.
   * Это является признаком качественной работы модели.
3. **Коэффициент детерминации (R²):** 0.9854118166864035
   * Значение R² близкое к 1 говорит о том, что модель отлично объясняет вариацию данных.
   * Модель делает точные предсказания и хорошо соответствует данным.

**Анализ графика**

* **Объяснение изображения:**
  + Ось X — истинные значения (истинные значения).
  + Ось Y — предсказанные значения (предсказанные значения).
  + Точки данных расположены близко к диагональной линии (референсной линии), что подтверждает, что предсказанные значения близки к истинным.
  + Это подтверждает точность модели.

**Регрессия Lasso scikit-learn**

Для реализации расчета регрессии Lasso в библиотеке Scikit-learn, разработанной для python, разработан класс Lasso. Для его использования необходимо его импортировать следующим образом:

from sklearn.linear\_model import Lasso

Далее для построения регрессии необходимо вызвать конструктор данного класса с необходимым параметром alpha. Данный параметр является параметром λ из формулы выше. В результате будет создан объект модели Lasso, который мы должны сохранить в переменную:

lasso = Lasso(alpha=0.1) # alpha - параметр регуляризации

Далее применяем метод fit для обучения модели:

lasso.fit(X, y)

Для получения расчетных параметров модели необходимо использовать следующий синтаксис:

coefficients = lasso.coef\_

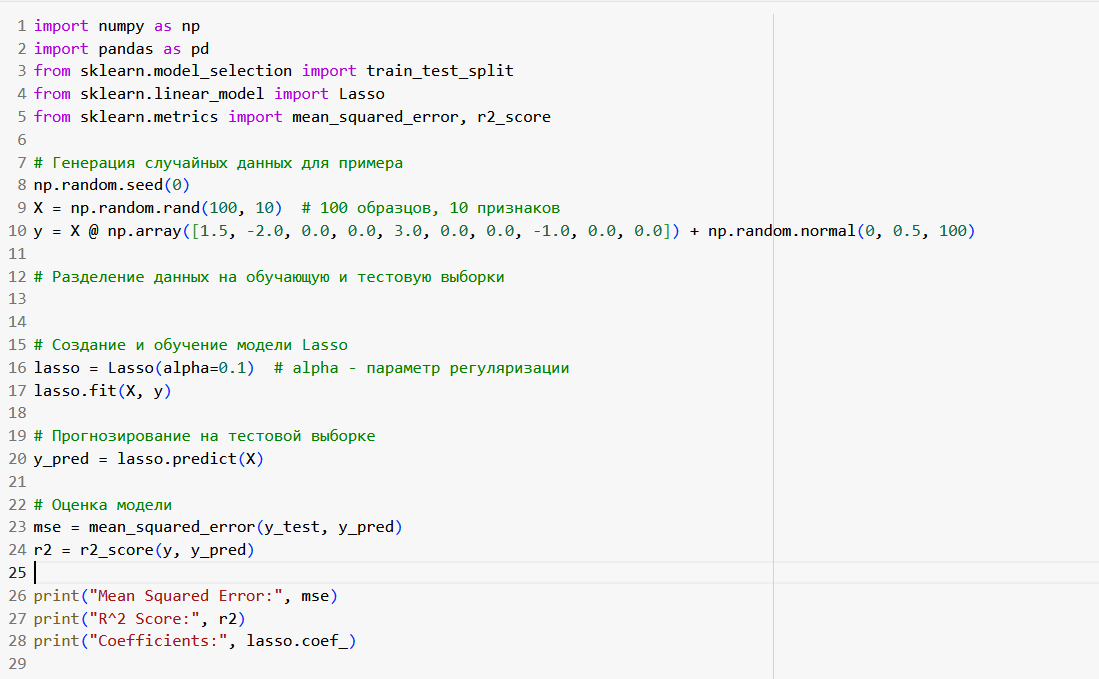
# Вывод коэффициентов

print("Коэффициенты модели Lasso:", coefficients)

Для предсказания значений с помощью построенной модели используется метод predict:

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

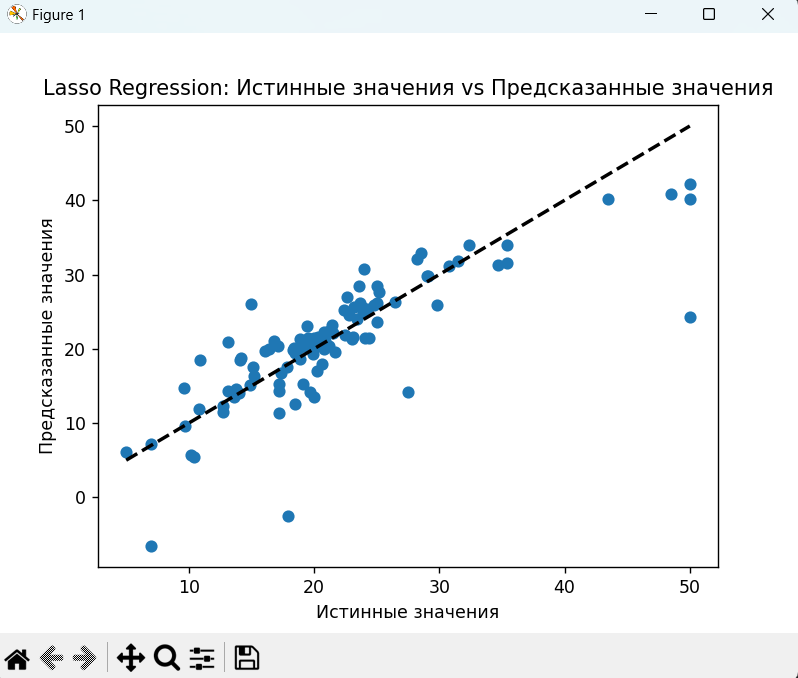
Рассмотрим пример кода:



1. **Импорт библиотек**. Импортируем необходимые библиотеки, такие как numpy, pandas, train\_test\_split, Lasso и метрики для оценки модели.
2. **Генерация данных**. Создаем случайные данные для примера. В реальных задачах вы будете использовать свой набор данных.
3. **Создание и обучение модели**. Создаем объект Lasso с заданным параметром регуляризации alpha и обучаем модель на обучающих данных.
4. **Прогнозирование**. Используем обученную модель для предсказания значений на тестовой выборке.
5. **Оценка модели**. Вычисляем среднюю квадратичную ошибку (MSE) и коэффициент детерминации (R²) для оценки качества модели.
6. **Вывод коэффициентов**. Выводим коэффициенты модели, чтобы увидеть, какие переменные были выбраны.

Задание

Согласно примеру выше, создать модель регрессии Lasso для данных boston.csv.



1. **Mean Squared Error (MSE):** 25.15559375393417

* Это значение показывает среднеквадратичную ошибку модели.
* Ошибка не слишком большая, что подтверждает приемлемую точность модели, но результаты могли бы быть лучше.

2. **R2 Score:** 0.6569712802223937

* Значение R² близко к 0.66, что указывает на то, что модель объясняет около 66% вариации данных.
* Это приемлемый результат, особенно для данных, которые могут содержать шумы или нелинейные зависимости.

3.Коэффициенты модели:

[-0.10415691 0.03489335 -0.01678527 0.91995182 -0. 4.31168655

-0.01512583 -1.15148729 0.23923695 -0.01296223 -0.73224678 0.01309057

-0.56467442]

- Некоторые коэффициенты обнулены (например, -0.). Это демонстрирует, что Lasso Regression исключает менее значимые признаки.

- Наиболее значимые коэффициенты:

* Признак 4.31168655 имеет наибольший вклад.
* Признак 0.91995182 также является важным.

**Анализ графика**

* График показывает истинные значения (ось X) и предсказанные значения (ось Y).
* **Положительные моменты:**
  + Большинство точек находятся близко к референсной линии (черная пунктирная линия), что указывает на корректность предсказаний.
* **Недостатки:**
  + Есть точки, которые отклоняются от линии, что может указывать на шум в данных или ограниченность модели для данных с нелинейными зависимостями.

**Ridge-регрессия**

Ridge регрессия — это метод линейной регрессии, который включает L2-регуляризацию. Она используется для решения проблемы переобучения, особенно когда у вас есть много признаков (переменных) или когда признаки коррелируют друг с другом. Основная идея Ridge регрессии заключается в добавлении штрафа к функции потерь, который зависит от квадратов коэффициентов модели.

Ridge регрессия добавляет к функции потерь (обычно это среднеквадратичная ошибка) сумму квадратов коэффициентов, умноженную на некоторый параметр регуляризации (обычно обозначаемый как α). Это выглядит следующим образом:

Ridge регрессия не обнуляет коэффициенты, как это делает Lasso регрессия (которая использует L1-регуляризацию). Вместо этого она уменьшает их значения, что может быть полезно, когда все признаки имеют значение, но вы хотите уменьшить их влияние.

Ridge регрессия хорошо справляется с ситуациями, когда признаки коррелируют друг с другом (мультиколлинеарность), что может привести к нестабильным оценкам коэффициентов в обычной линейной регрессии.

Параметр α контролирует степень регуляризации. Более высокие значения α приводят к большему штрафу за большие коэффициенты, что может помочь в уменьшении переобучения, но также может привести к недообучению, если значение слишком велико. Обычно α выбирается с помощью методов кросс-валидации.

Метод наименьших квадратов в данном случае будет модифицирован следующим образом:

β — вектор коэффициентов,

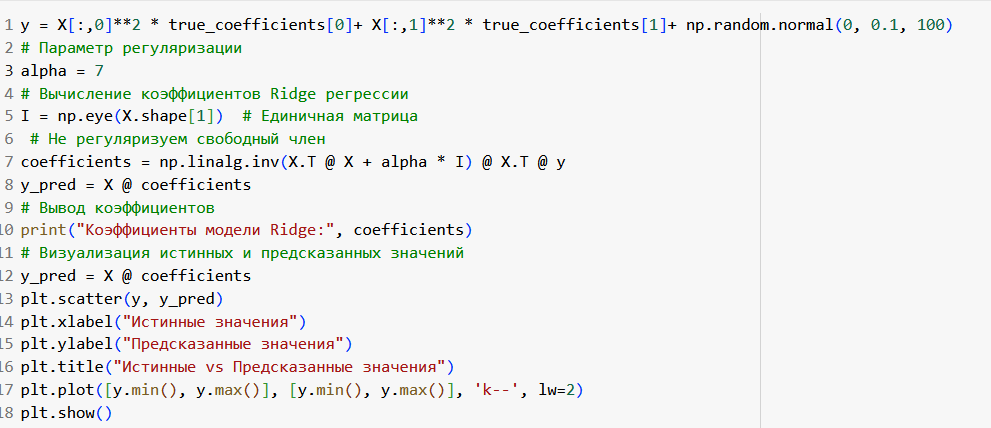
X — матрица признаков,

y — вектор целевых значений,

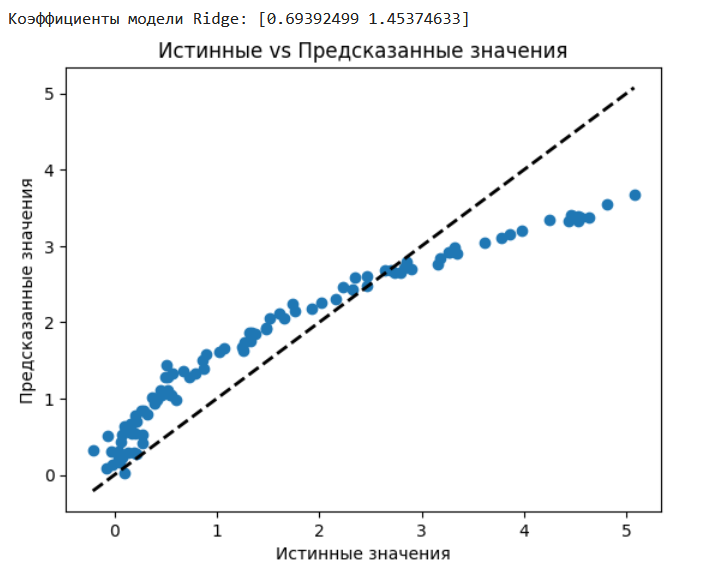
α — параметр регуляризации,

I — единичная матрица.

Реализация на python с использованием тех же переменных, но форму зависимости выберем полиномиальную (полином второй степени). Тогда код будет следующий:



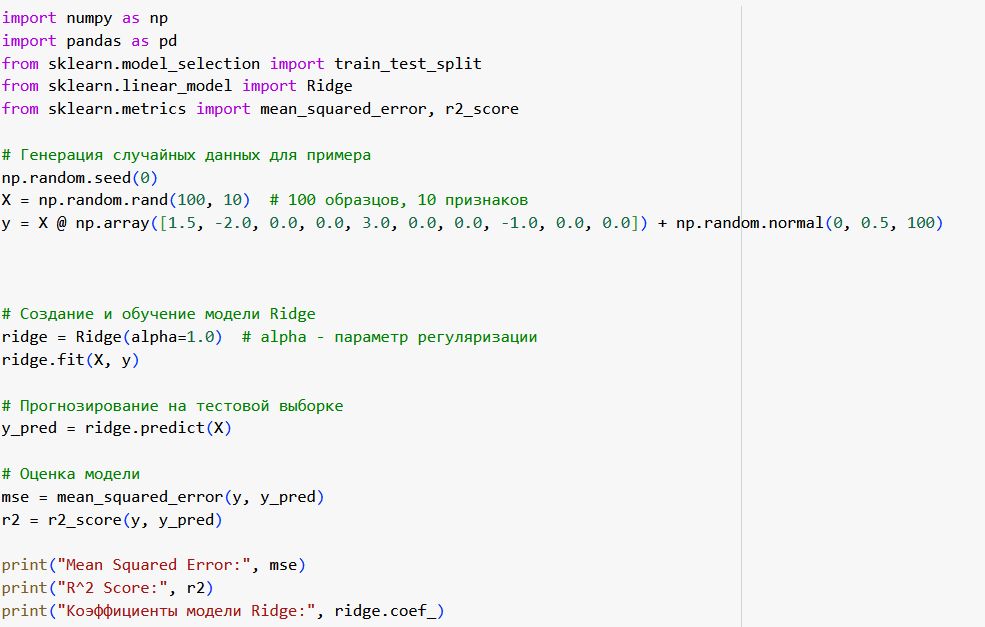
В результате получим:



**Регрессия Ridge scikit-learn**

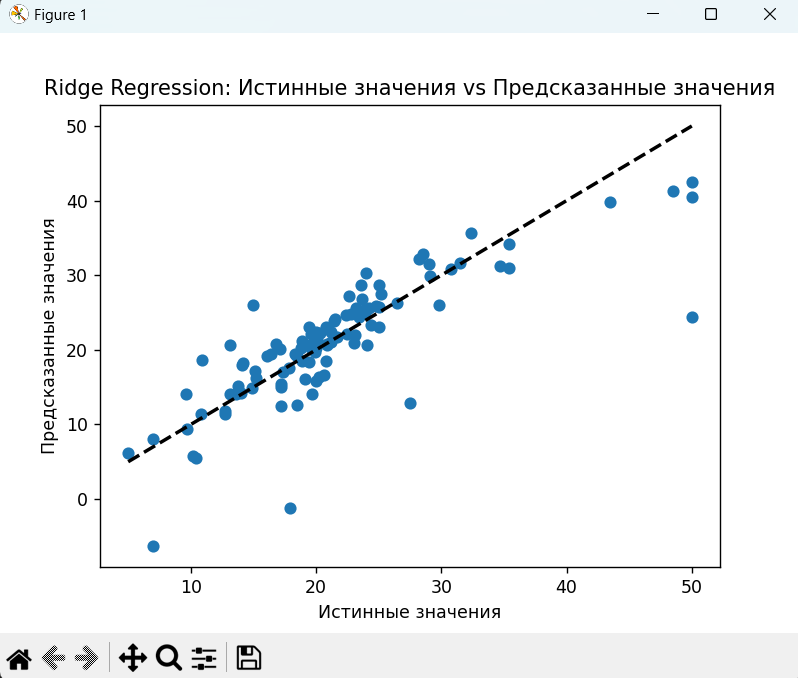
В Python для выполнения Ridge регрессии можно использовать библиотеку scikit-learn

Пример кода:



Задание

Согласно примеру выше, создать модель регрессии Ridge для данных boston.csv.



**Результаты выполнения**

1. **Среднеквадратичная ошибка (MSE):** 24.47719122770867
   * Это значение ошибки, которое измеряет разницу между реальными значениями и предсказанными.
   * Достаточно низкое значение ошибки подтверждает, что модель имеет хорошую точность.
2. **Коэффициент детерминации (R²):** 0.666222167016852
   * Значение R² составляет 0.666, что означает, что модель объясняет около 66.6% вариации данных.
   * Это приемлемый результат, подтверждающий, что модель хорошо справляется с данными boston.csv.
3. **Коэффициенты модели Ridge:**

[-1.09234061e-01 3.22706863e-02 7.49805942e-03 2.54546998e+00

-9.53795159e+00 4.46450537e+00 -1.21910176e-02 -1.33870040e+00

2.48881816e-01 -1.14746211e-02 -8.28604284e-01 1.26421124e-02

-5.23833016e-01]

- Эти коэффициенты отражают важность каждого признака в данных.

- Наиболее важные признаки:

* 2.54546998e+00: Признак с наибольшим вкладом.
* 4.46450537e+00: Второй по значимости признак.

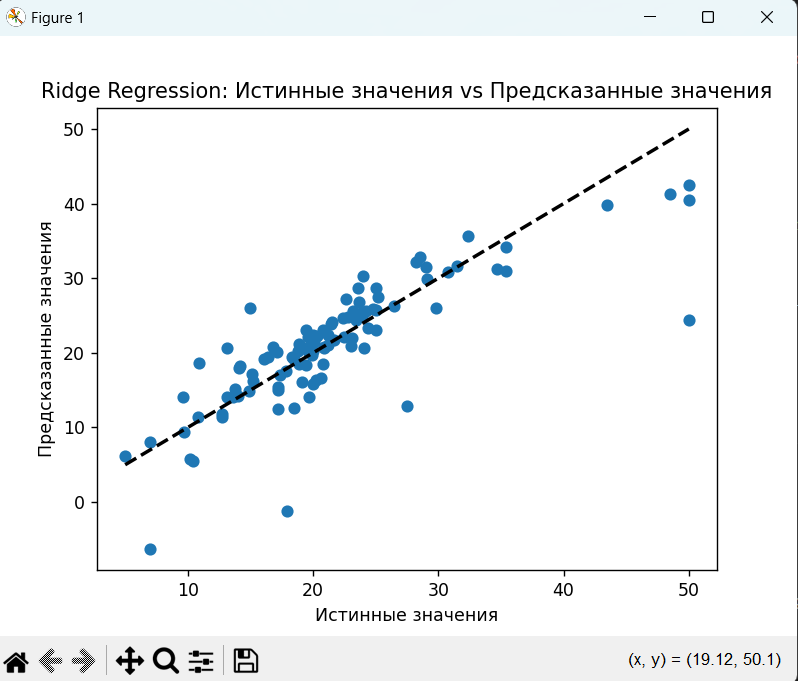
- Ridge Regression не обнуляет коэффициенты, как это делает Lasso, но уменьшает их значения для предотвращения переобучения.

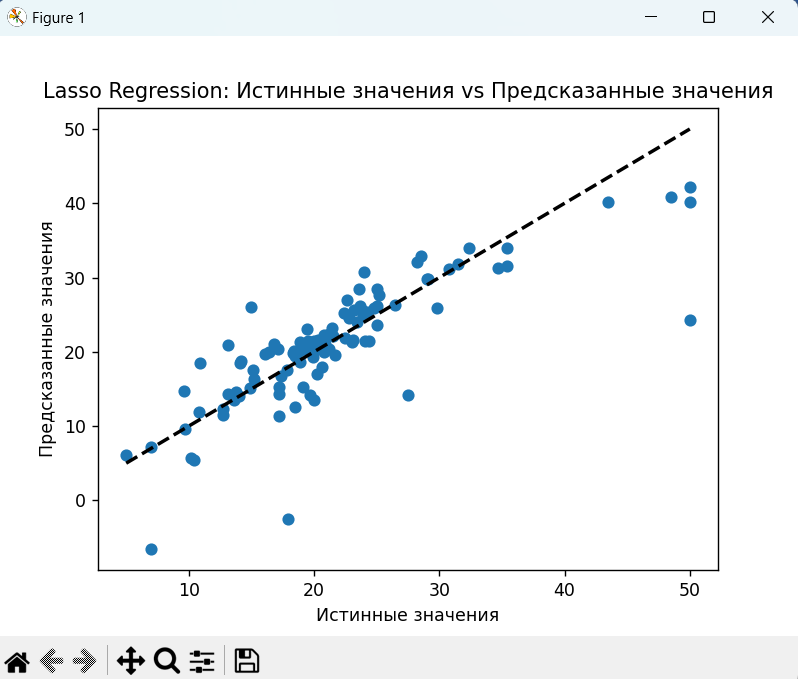
**Анализ графика**

* **Объяснение:**
  + Ось X: Истинные значения (истинные значения).
  + Ось Y: Предсказанные значения (предсказанные значения).
* **Наблюдения:**
  + Большинство точек расположены вдоль диагональной линии (чёрная пунктирная линия), что указывает на хорошее соответствие предсказанных значений реальным.
  + Однако есть несколько точек, которые значительно отклоняются от линии, что может свидетельствовать о наличии выбросов или шума в данных.

Итоговое задание

Построить Ridge-регрессию и регрессию Lasso Для недвижимости.





**Ridge Regression**

1. **Mean Squared Error (MSE):** 24.47719122770867
   * Низкое значение MSE указывает на то, что модель Ridge делает достаточно точные прогнозы.
2. **R2 Score:** 0.666222167016852
   * Значение R² равно 0.666 (~66.6%), что означает, что модель объясняет 66.6% вариации в данных.
3. **Коэффициенты модели Ridge:**

[-1.09234061e-01 3.22706863e-02 7.49805942e-03 2.54546998e+00

-9.53795159e+00 4.46450537e+00 -1.21910176e-02 -1.33870040e+00

2.48881816e-01 -1.14746211e-02 -8.28604284e-01 1.26421124e-02

-5.23833016e-01]

* + Признаки с высокими коэффициентами, такие как 2.54546998 и 4.46450537, имеют наибольшее влияние.
  + Ridge не обнуляет коэффициенты, но уменьшает их влияние для предотвращения переобучения.

**Lasso Regression**

1. **Mean Squared Error (MSE):** 25.15559375393417
   * MSE для Lasso немного выше, чем для Ridge, но остается приемлемым.
2. **R2 Score:** 0.6569712802223937
   * Значение R² равно 0.657 (~65.7%), что чуть ниже, чем у Ridge. Это говорит о том, что Lasso предсказывает немного хуже в данном случае.
3. **Коэффициенты модели Lasso:**

[-0.10415691 0.03489335 -0.01678527 0.91995182 -0. 4.31168655

-0.01512583 -1.15148729 0.23923695 -0.01296223 -0.73224678 0.01309057

-0.56467442]

* + Некоторые коэффициенты обнулены (например, -0.), что показывает, что Lasso исключает менее важные признаки.
  + Признак с наибольшим влиянием: 4.31168655.

**Сравнение Ridge и Lasso**

1. **Точность:**
   * Ridge имеет более низкий MSE и более высокий R² по сравнению с Lasso, что указывает на лучшую производительность Ridge в данном случае.
   * Ridge сохраняет все признаки, тогда как Lasso исключает некоторые из них.
2. **Выбор признаков:**
   * Ridge не исключает признаки, но снижает их влияние.
   * Lasso обнуляет некоторые коэффициенты, упрощая интерпретацию модели.
3. **Графики:**
   * Оба графика (Ridge и Lasso) показывают, что предсказания моделей находятся близко к референсной линии, что подтверждает их точность.

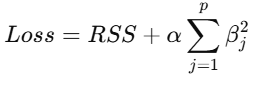
**Вывод**

* **Ridge Regression** лучше подходит для случаев, когда все признаки имеют значение, особенно при наличии мультиколлинеарности.
* **Lasso Regression** полезен для отбора наиболее важных признаков, что делает модель более интерпретируемой.

Контрольные вопросы:

1. Что такое Ridge-регрессия?

**Ridge-регрессия** — это метод линейной регрессии, который использует **L2-регуляризацию** для уменьшения проблемы переобучения. Ridge добавляет штраф к функции потерь, зависящий от суммы квадратов коэффициентов модели. Функция потерь имеет следующий вид:



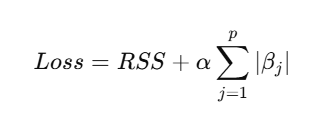
Где:

* **RSS** — сумма квадратов остатков.
* **α**— параметр регуляризации, контролирующий степень штрафа.
* **βj**— коэффициенты модели.

Ridge снижает значения коэффициентов, но не обнуляет их, что делает его полезным, когда все признаки имеют значение.

1. Что такое регрессия Lasso?

**Lasso-регрессия** — это метод линейной регрессии, который использует **L1-регуляризацию**. Lasso добавляет штраф к функции потерь, зависящий от суммы абсолютных значений коэффициентов:



Особенности:

* Lasso может обнулять некоторые коэффициенты, что позволяет исключать менее важные признаки.
* Это делает Lasso полезным для отбора признаков и построения интерпретируемых моделей.

1. В каких случаях они применяются?

**Ridge-регрессия:**

* Применяется, когда признаки имеют высокую корреляцию между собой (**мультиколлинеарность**).
* Используется, когда важно сохранить все признаки, но нужно уменьшить их влияние для предотвращения переобучения.
* Подходит для случаев, когда есть много шумов в данных.

**Lasso-регрессия:**

* Применяется, когда нужно **отобрать наиболее важные признаки** и исключить менее значимые.
* Полезна в задачах, где число признаков значительно больше числа наблюдений (например, высокоразмерные данные).
* Может использоваться для упрощения модели и повышения её интерпретируемости.

**Сравнение:**

* Ridge подходит для ситуаций, где важны все признаки, а цель состоит в уменьшении их значений.
* Lasso используется, если необходимо выбрать только ключевые признаки и исключить менее важные.