Харківський національний університет радіоелектроніки

(повне найменування вищого навчального закладу)

Кафедра штучного інтелекту

(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Машинне навчання»

(назва дисципліни)

на тему: практичне дослідження узагальнених лінійних моделей (Generalized Linear Models) з використанням інструментарію Scikit-learn на мові програмування Python

Студента (ки) _3_ курсу 17-7 групи
напряму підготовки_____ІТКН_____
__Нефьодова Даниїла Андрійовича____
(прізвище та ініціали)

Керівник _доцент каф. ШІ, доц., к.т.н.___
_____ Вітько О. В._____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____Оцінка: ECTS _____

Члени комісії _____ ____Вітько О. В._____
(підпис) (прізвище та ініціали)
_____ ____Кулішова Н.Є.___
(підпис) (прізвище та ініціали)
_____ _____Філатов В.О._____
(підпис) (прізвище та ініціали

Харків - 2019

Харківський національний університет радіоелектроніки
Інститут, факультет, відділення КН
Кафедра, циклова комісія ШІ
Освітньо кваліфікаційний рівень бакалавр
Напрям підготовки 6.050101 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
_____
"____" _____20__ року

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ
З ДИСЦИПЛІНИ «МАШИННЕ НАВЧАННЯ»

студенту Нефьодову Даниїлу Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: практичне дослідження узагальнених лінійних моделей (Generalized Linear Models) з використанням інструментарію Scikit-learn на мові програмування Python

2. Термін здачі студентом закінченої роботи 26.12.2019

3. Вихідні дані до проекту: алгоритми з пакету sklearn.linear_models, наукові джерела, статті, мова програмування Python 3.7 та надбудова над нею IPython, середа для аналізу данних Jupyter Lab, пакет Anaconda.

4. Зміст розрахунково-пояснювальної записки (перелік питань, котрі під лягають розробці): опис предметної області, аналіз даних, перевірка методів узагальнених лінійних моделей, алгоритм лінійної регресії, ridge регресії, регресії lasso, elastic net логістичної регресії.

5. Дата видачі завдання: 20.10.2019

Керівник роботи Вітько Олександра Валеріївна

(підпис) (прізвище, ім’я, по батькові)

Студент Нефьодов Даниїл Андрійович

(підпис) (прізвище, ім’я, по батькові)

«___» _____20__р

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів курсового проекту	Термін виконання	Примітка
1	Отримання завдання на курсову роботу	22.10.2019	виконано
2	Аналіз завдання	23.10.2019-6.10.2019	виконано
3	Пошук набору данних	27.10.2019	виконано
4	Попередній аналіз данних	01.11.2019-8.11.2019	виконано
5	Визначення вихідних алгоритмів	19.11.2019-5.11.2019	виконано
6	Написання програм для обраних моделей	27.11.2019	виконано
7	Оформлення програмної частини проекту	28.11.2019-0.11.2019	виконано
8	Оформлення пояснювальної записки	02.12.2019-3.12.2019	виконано
9	Розробка презентації, підготовка доповіді	15.12.2019	виконано
10	Захист курсової роботи	26.12.2019	

Керівник роботи_____ Вітько Олександра Валеріївна _

(підпис) (прізвище, ім’я, по батькові)

Студент _____ Нефьодов Даниїл Андрійович __

(підпис) (прізвище, ім’я, по батькові)

«___» _____20__р

РЕФЕРАТ

Пояснювальна записка до міждисциплінарного курсового проекту містить 27 сторінок, 8 рисунків, 18 формул, 2 джерела.

PYTHON, SCIKIT-LEARN, DATASET, CORRELATION, OUTLIERS, MACHINE LEARNING, ALGORITHM, GENERALIZED LINEAR MODELS, LINEAR REGRESSION, RIDGE REGRESSION, LASSO REGRESSION, LOGISTIC REGRESSION, CLASSIFICATION.

Об'єктом досліджень міждисциплінарного курсового проекту є узагальнені лінійні моделі пакету з пакету sklearn. linear_models.

Предметом досліджень курсового проекту є програмно згенеровані датасети, на яких досліджуються узагальнені лінійні моделі.

Мета досліджень: дослідження алгоритмів і розробка програмних засобів з використанням узагальнених лінійний моделей.

В роботі проведено аналіз предметної області, що відноситься до регресійного аналізу даних. За результатами експериментів проведено аналіз відповідності розробленого програмного забезпечення висунутим вимогам.

Дослідницький аналіз даних, підготовка та візуалізація даних проводився за допомогою бібліотек sklearn, Pandas, numpy та matplotlib мови програмування Python.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів

GLM – узагальнені лінійні моделі (з англ. Generalized Linear Models).

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.

scikit-learn, sklearn – бібліотека мови програмування Python для аналізу даних.

Мультіколлінеарність – наявність лінійної залежності між двома або більше факторними змінними у регресійній моделі.

ВСТУП

На сьогодні жодна сфера життя суспільства не може обійтись без прогнозів як засобу передбачення поведінки процесів та об’єктів у майбутньому.

Одним за таких засобів є регресійний аналіз, моделі якого склали йому репутація надійного інструменту аналізу. Головним плюсом цього методу є зведення причини і наслідку та простота реалізації більшості моделей.

У даній роботі буде розглянуто набір моделей з пакету sklearn.linear_models.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про узагальнену лінійну модель

Узагальнена лінійна модель – це статистична лінійна модель, що визначається наступним рівнянням:

$$Y = WX + W_0, \qquad (1.1)$$

де Y – це залежна передбачувана величина, X – незалежна передбачувана величина, W – вектор коефіцієнтів, W_0 – вектор перетину.

Але лінійні моделі підходять тільки для даних, що задовольняють наступним умовам:

1) Лінійний взаємозв’язок між змінними;
2) Нормальний розподіл залишків – різниця між реальними даними та даними на регресійній прямій;
3) Перевірка на мультіколлінеарність;
4) Бажано нормальне розподілення змінних.

До узагальнених лінійних моделей відносять:

1) Метод найменших квадратів (Ordinary Least Squares);
2) Ridge-регресію;
3) Регресію Lasso;
4) Elastic Net;
5) Логістичну регресію.

1.2 Постановка задачі на курсову роботу

Головним завданням курсового проектування є дослідження вищевказаних моделей. Розробка програм, що засновані на даних моделях. Дослідження принципів та умов їх застосування.

На базі створених моделей, ознайомитися з їх поведінкою на згенерованих вибірках даних. Проаналізувати отримані результати.

2 ОПИС АЛГОРИТМІВ УЗАГАЛЬНЕНИХ ЛІНІЙНИХ МОДЕЛЕЙ

2.1 Лінійна регресія

У статистиці лінійна регресія — це метод моделювання залежності між залежною змінною Y та незалежною змінною X.

В основі мінімізації функції втрат лінійної регресії є ідея метода найменших квадратів, що визначає W та W_0 для того, щоб використовувати отриману модель робити подальші прогнози.

Для спрощення використаємо вектор формулу з одним предикатом:

$$y_i = a + bx_i. \qquad (2.1)$$

За методом найменшим квадратів ми мінімізуємо квадратичні помилки:

$$\hat{a} = \min_a \sum_{i=1}^{n} (y_i - a - bx_i)^2 = \min_a \sum_{i=1}^{n} {\varepsilon_i}^2; \qquad (2.2)$$

$$\hat{b} = \min_b \sum_{i=1}^{n} (y_i - a - bx_i)^2 = \min_b \sum_{i=1}^{n} {\varepsilon_i}^2. \qquad (2.3)$$

Визначимо квадратичну суму помилок як:

$$S(\hat{a}, \hat{b}) = \sum_{i=1}^{n} (y_i - a - bx_i)^2. \qquad (2.4)$$

Тоді зведемо задачу оптимізації наступним чином:

$$\hat{b}: \frac{\delta S(\hat{a}, \hat{b})}{\delta \hat{b}} = 0; \qquad (2.5)$$

$$\hat{a}: \frac{\delta S(\hat{a}, \hat{b})}{\delta \hat{a}} = 0. \qquad (2.6)$$

Перетворивши вирази отримаємо:

$$\hat{b} = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad (2.7)$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x} \qquad (2.8)$$

Отримавши значення $\hat{a}$ та $\hat{b}$ отримаємо рівняння моделі, яке буде виглядати як:

$$\bar{y} = \hat{a} + \hat{b}\bar{x} \qquad (2.9)$$

2.2 Ridge-регресія

У випадку Ridge-регресії до формули 2.2 додається штраф, еквівалентний квадрату суми коефіцієнтів вектора W:

$$\hat{b} = \min_b \sum_{i=1}^{n} (y_i - a - bx_i)^2 + \lambda b^2, \qquad (2.10)$$

де $\lambda > 0$ – параметр складності, який контролює коефіцієнти нахилу.

При $\lambda \to 0$ графік регресії наближається до графіка аналогічної функції вартості лінійної регресії.

Ridge підходить у випадках, якщо існує мало предикатів та усі вони мають бути релевантними для передбачення.

2.3 Регресія Lasso

У випадку регресії Lasso до формули 2.2 додається штраф, але вже еквівалентний сумі модулів коефіцієнтів вектора W:

$$\hat{b} = \min_b \sum_{i=1}^{n} (y_i - a - bx_i)^2 + \lambda |b|, \qquad (2.11)$$

де $\lambda > 0$ – параметр складності, який контролює коефіцієнти нахилу.

Точно так як і функція вартості Ridge-регресії п ри $\lambda \to 0$ графік регресії наближається до графіка аналогічної функції вартості лінійної регресії.

Алгоритм Lasso використовується, якщо мається багато предикатів, деякі з яких не такі важливі, ніж інші.

2.4 Elastic Net

Elastic Net – це гібрид Lasso та Ridge, де включені штрафи як по абсолютній величині, так і у квадраті, що регулюються коефіцієнтом r є $\overline{0,1}$:

$$\hat{b} = \min_b \sum_{i=1}^{n} (y_i - a - bx_i)^2 + \lambda r|b| + \frac{\lambda(1-r)}{2} b^2, \qquad (2.12)$$

де $\lambda > 0$ – параметр складності, який контролює коефіцієнти нахилу.

Головна перевага Elastic Net над Lasso у тому, що у ньому знімається обмеження на мінімальну кількість записів, для видачі адекватного рішення системи.

2.5 Логістична регресія

Логістична регресія - це статистичний метод аналізу даних, в якому є один або кілька незалежних значень, що визначають результат. Результат вимірюється за допомогою дихотомічної змінної – у котрій є тільки два можливих варіанта або класа 0, або класа 1.

Тож по-перше потрібно розрахувати ймовірність того, що спостереження класу 1:

$$P(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + \cdots + w_n x_n)}}. \qquad (2.13)$$

Коефіцієнти $w_0, w_1 \ldots w_n$ обрані так, щоб максимізувати вірогідність приналежності спостереження до класу 1.

Після чого обирається порогова границя C, яка чітко класифікує задане вхідне значення в один з класів, яка породжує та визначає ступінь прийнятності до помилок 1-го та 2-го родів.

2.6 Використані метрики

Для оцінки результатів моделювання лінійної регресії, Ridge, Lasso та Elastic Net викостаємо метрики середньоквадратичної помилки та коефіцієнт детермінації.

Середньоквадратична помилка (позначається як $RMSE$)– величина, що характеризує стандартне відхилення вибіркового середнього, розраховане по вибірці розміром n n із генеральної сукупності.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}{n}}. \qquad (2.14)$$

Коефіцієнт детермінації (позначається як $R2$) — статистичний показник, що використовується в статистичних моделях як міра залежності варіації залежної змінної від варіації незалежних змінних. Вказує наскільки отримані спостереження підтверджують модель.

$$R2 = \frac{\sum_{i=1}^{n}(\widehat{y_i} - \bar{y})^2}{\sum_{i=1}^{n}(\widehat{y_i} - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \bar{y})^2}. \qquad (2.15)$$

Частка правильно класифікованих об’єктів (позначається як $Accuracy$) — ймовірність того, що клас передбаченої правильно.

$$Accuracy(a) = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (2.16)$$

де TP, TN, FP, FN – кількості правильних позитивних, правильних негативних, хибних позитивних та хибних негативних прогнозів відповідно.

F-міра – це поєднання метрик точності та повноти

$$F_{measure} = \frac{2Presicion * Recall}{Presicion + Recall}. \qquad (2.17)$$

3 ОПИС ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

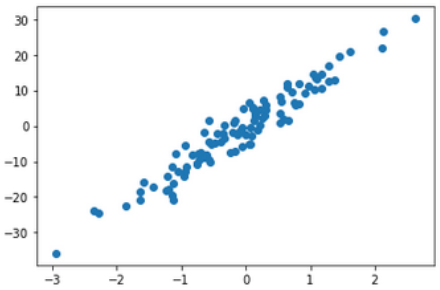
3.1 Вибір датасетів

По-перше потрібно згенерувати дата сет. Scikit-learn містить велику кількість генераторів датасетів, генерацію яких можна контролювати вхідними параметрами функцій генерування. Ми використаємо make_regression() для генерування датасета для лінійної регресії та методів її регуляризації та make_classification() для логістичної регресії.

```
In [431]: from sklearn.datasets import make_regression

          X, y = make_regression(n_samples=100, n_features=1,
                                 n_informative=10, n_targets=1,
                                 bias=0.0, tail_strength=0.5,
                                 noise=3)
          plt.scatter(X, y)

Out[431]: <matplotlib.collections.PathCollection at 0x1e9fbd6bcc8>
```




```
In [93]: x, y = make_classification(
             n_samples=100,
             n_features=1,
             n_classes=2,
             n_clusters_per_class=1,
             flip_y=0.03,
             n_informative=1,
             n_redundant=0,
             n_repeated=0
         )
         plt.scatter(x, y, c=y, cmap='rainbow')

Out[93]: <matplotlib.collections.PathCollection at 0x263d0079248>
```

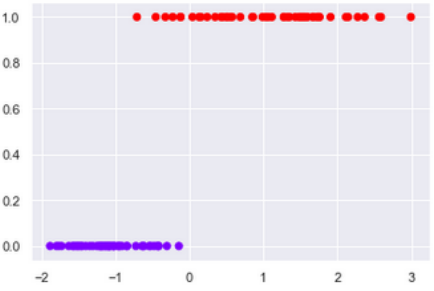


Рисунок 3.1 – Фрагмент коду та змодельований датасет

3.2 Програмування та порівняння результатів та оцінок лінійної регресії, Ridge, Lasso та Elastic Net

Після генерації датасета, на ньому перевіримо дані методи. Порівнювати методи будемо за метриками RMSE та R2:

```
The model performance for training set
--------------------------------------
RMSE is 2.851340886683267
R2 score is 0.9269381119216248


The model performance for testing set
--------------------------------------
RMSE is 3.5811997726447866
R2 score is 0.9232705718046434
```

Рисунок 3.2 – Результат лінійної регресії

```
The model performance for training set
--------------------------------------
RMSE is 5.823829126700233
R2 score is 0.6952035839412184


The model performance for testing set
--------------------------------------
RMSE is 7.988520864836814
R2 score is 0.6181985152520832
```

```
The model performance for training set
--------------------------------------
RMSE is 2.854418827312087
R2 score is 0.9965016541110492


The model performance for testing set
--------------------------------------
RMSE is 3.3046754602225104
R2 score is 0.9952293872053899
```

```
The model performance for training set
--------------------------------------
RMSE is 7.346660293308864
R2 score is 0.5149656177342359


The model performance for testing set
--------------------------------------
RMSE is 9.86931294510048
R2 score is 0.4172547991068969
```

```
The model performance for training set
--------------------------------------
RMSE is 10.548815488535352
R2 score is 1.8538737212026035e-10


The model performance for testing set
--------------------------------------
RMSE is 13.764507055428632
R2 score is -0.13351217892997091
```

Рисунок 3.3 – Результат методу Ridge з параметром λ,

що дорівнює 1, $\rightarrow$ 0, 2, 10^{10}

```
The model performance for training set
--------------------------------------
RMSE is 8.254098669877393
R2 score is 0.3877455926658997

The model performance for testing set
--------------------------------------
RMSE is 10.978141730010222
R2 score is 0.27895473062972087
```

```
The model performance for training set
--------------------------------------
RMSE is 77.51212901767097
R2 score is -52.992313813650846

The model performance for testing set
--------------------------------------
RMSE is 92.13838895495863
R2 score is -49.790921685503534
```

```
The model performance for training set
--------------------------------------
RMSE is 10.54881548951316
R2 score is 0.0

The model performance for testing set
--------------------------------------
RMSE is 13.764507056612635
R2 score is -0.13351217912497693
```

```
The model performance for training set
--------------------------------------
RMSE is 10.54881548951316
R2 score is 0.0

The model performance for testing set
--------------------------------------
RMSE is 13.764507056612635
R2 score is -0.13351217912497693
```

Рисунок 3.4 – Результат методу Lasso з параметром λ,

що дорівнює $1, \rightarrow 0, 2, 10^{10}$

```
The model performance for training set
--------------------------------------
RMSE is 4.320061798562518
R2 score is 0.8322847605059233

The model performance for testing set
--------------------------------------
RMSE is 6.066671443767511
R2 score is 0.7798058301976761

The model performance for training set
--------------------------------------
RMSE is 5.873398430518402
R2 score is 0.68999297637256

The model performance for testing set
--------------------------------------
RMSE is 8.050375224083345
R2 score is 0.6122631199124058
```

```
The model performance for training set
--------------------------------------
RMSE is 2.85134429460453
R2 score is 0.9269379372744531

The model performance for testing set
--------------------------------------
RMSE is 3.582859559602738
R2 score is 0.9231994313908489

The model performance for training set
--------------------------------------
RMSE is 10.54881548951316
R2 score is 0.0

The model performance for testing set
--------------------------------------
RMSE is 13.764507056612635
R2 score is -0.13351217912497693
```

Рисунок 3.5 – Результат методу Elastic Net з параметром λ,

що дорівнює $1, \rightarrow 0, 2, 10^{10}$

Отримані результати свідчать, що на даному дата сеті най адекватніше спрацьовує лінійна регресія, з показником $R2 \approx 0.923$, інші моделі видають гірший результат, що збільшується при зменшенні параметра λ, тобто регулярізація у даному датасеті зайва, адже не виконується умова перенавчання при використанні мінімізації найменшої суми квадратів у лінійній регресії.

3.3 Програмування та оцінка результатів логістичної регресії

Порівняємо результати навчання логістичної регресії при різних порогових границях C. Для цього візьмемо значення C, що дорівнюють 1, 0.5 та 0.1. У ході проведення дослідів ми отримали наступні дані для вище перелічених C відповідно:

```
In [113]: y_pred1 = lr1.predict(x_test)
          print(y_pred1)
          y_pred05 = lr05.predict(x_test)
          print(y_pred01)
          y_pred01 = lr01.predict(x_test)
          print(y_pred01)

          [1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 0 0 1]
          [1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1 0 0 1]
          [1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1 0 0 1]
```

Рисунок 3.7 – Результати прогнозування моделей

```
In [118]: print(accuracy_score(y_test, y_pred1))
          print(accuracy_score(y_test, y_pred05))
          print(accuracy_score(y_test, y_pred01))

          1.0
          1.0
          0.96
```

```
In [119]: print(f1_score(y_test, y_pred1))
          print(f1_score(y_test, y_pred05))
          print(f1_score(y_test, y_pred01))

          1.0
          1.0
          0.967741935483871
```

Рисунок 3.8 – Оцінки моделей за метриками частки правильно класифікованих об’єктів та F-мірою

Для коефіцієнтів 0,5 та 1 ми маємо точність та F міру дорівнюють 1, це показує нам що ці за цих значеннях порогової границі C для цього набору даних складена майже точна вихідна модель.

ВИСНОВКИ

У даній курсовій роботі ми теоретично та практично дослідили алгоритми узагальнених лінійних моделей з пакету sklearn. linear_models.

В ході дослідження були згенеровані декілька наборів даних, що підходять для перевірки робото спроможності кожного з наступних алгоритмів:

1) Лінійна регресія;
2) Ridge-регресія;
3) Регресія Lasso;
4) Elastic Net;
5) Логістична регресія.

Датасети були згенеровані за допомогою засобів бібліотек sklearn та numpy мови програмування Python.

В ході перевірки алгоритмів лінійної регресії та її регулярізацій, було виявлено що до згенерованих дата сетів, які мають нормальне розподілення даних та позитивну кореляцію стандартної лінійної регресії вистачає для побудови регресійної прямою, яка показує високий коефіцієнт детермінації, від 0.8 до 1 в залежності від параметрів регресії та самого детасету. З іншого боку методи регулярізації Ridge, Lasso та Elastic Net не показали настільки гарного результату, оскільки вони необхідні, в основному, для покращення роботи перенавчених моделей лінійної регресії та є дуже ситуативними.

Логістична регресія показало себе добре для згенерованого для неї дата сету, що було очікувано. Незважаючи на те, що датасет та навчальні моделі на ньому не стали складною задачею, були отримані корисні практичні, щодо реалізації та застосування логістичної регресії.

Також даних пакет містить один з головних методів оптимізації алгоритмів – стохастичний градієнтний спуск. Його реалізація поза межами моделі не є доцільною, оскільки він не покаже своїх переваг на згенерованому датасеті.

ДЖЕРЕЛА ПОСИЛАНЬ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.

2. Плас Дж. Вандер П37 Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018. — 576 с.: ил. — (Серия «Бестселлеры O’Reilly»). ISBN 978-5-496-03068-7

Міністерство освіти и науки України

Затверджую

Керівник курсового проекту,

_____

_____

(Підпис, дата, прізвище, ім'я, по батькові)

Практичне дослідження узагальнених лінійних моделей (Generalized Linear Models) з використанням інструментарію Scikit-learn на мові програмування Python

ДОДАТОК А

Текст програми

Студент групи ІТКН-17-7

(Назва групи)

Нефьодов Даниїл Андрійович

(Підпис, дата, прізвище, ім'я, по батькові)

2019

3МІСТ

КОД ПРОГРАМИ

1 predict_numbers.ipynb.

```
import numpy as np
from sklearn.datasets import make_regression

X, y = make_regression(n_samples=100, n_features=1,
                       n_informative=10, n_targets=1,
                       bias=0.0, tail_strength=0.5,
                       noise=3)
plt.scatter(X, y)
# Split data in train set and test set
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
random_state=5)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

from sklearn.linear_model import LinearRegression

lin_model = LinearRegression()
lin_model.fit(X_train, y_train)
from sklearn.linear_model import Ridge
```

```
ridge = Ridge(normalize = True)
ridge.fit(X_train, y_train)

ridge_0 = Ridge(alpha = 0.001, normalize = True)
ridge_0.fit(X_train, y_train)

ridge_2 = Ridge(alpha = 2, normalize = True)
ridge_2.fit(X_train, y_train)

ridge_10__10 = Ridge(alpha = 10**10, normalize = True)
ridge_10__10.fit(X_train, y_train)
from sklearn.linear_model import Lasso

lasso = Lasso(max_iter = 10000, normalize = True)
lasso.fit(X_train, y_train)

lasso_0 = Lasso(max_iter = 10000, alpha = -10, normalize = True)
lasso_0.fit(X_train, y_train)

lasso_2 = Lasso(max_iter = 10000, alpha = 2, normalize = True)
lasso_2.fit(X_train, y_train)

lasso_10__10 = Lasso(max_iter = 10000, alpha = 10**10, normalize = True)
lasso_10__10.fit(X_train, y_train)
from sklearn.linear_model import ElasticNet

enet = ElasticNet(l1_ratio=0.7)
enet.fit(X_train, y_train)
```

```
enet_0 = ElasticNet(alpha=0.001, l1_ratio=0.7)
enet_0.fit(X_train, y_train)

enet_2 = ElasticNet(alpha=2, l1_ratio=0.7)
enet_2.fit(X_train, y_train)

enet_10__10 = ElasticNet(alpha=10**10, l1_ratio=0.7)
enet_10__10.fit(X_train, y_train)
```

2 logistic.ipynb.

```
from sklearn.datasets import make_classification
from matplotlib import pyplot as plt
from sklearn.linear_model import LogisticRegression
import seaborn as sns
sns.set()
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import pandas as pd

x, y = make_classification(
    n_samples=100,
    n_features=1,
    n_classes=2,
    n_clusters_per_class=1,
    flip_y=0.03,
    n_informative=1,
```

```
    n_redundant=0,
    n_repeated=0
)
plt.scatter(x, y, c=y, cmap='rainbow')

x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)

lr1 = LogisticRegression(C=1)
lr1.fit(x_train, y_train)
lr05 = LogisticRegression(C=0.5)
lr05.fit(x_train, y_train)
lr01 = LogisticRegression(C=0.1)
lr01.fit(x_train, y_train)

y_pred1 = lr1.predict(x_test)
print(y_pred1)
y_pred05 = lr05.predict(x_test)
print(y_pred01)
y_pred01 = lr01.predict(x_test)
print(y_pred01)

from sklearn.metrics import accuracy_score, f1_score

print(accuracy_score(y_test, y_pred1))
print(accuracy_score(y_test, y_pred05))
print(accuracy_score(y_test, y_pred01))

print(f1_score(y_test, y_pred1))
```

```
print(f1_score(y_test, y_pred05))
print(f1_score(y_test, y_pred01))

df = pd.DataFrame({'x': x_test[:,0], 'y': y_test})
df = df.sort_values(by='x')
from scipy.special import expit
sigmoid_function = expit(df['x'] * lr.coef_[0][0] + lr.intercept_[0]).ravel()
plt.plot(df['x'], sigmoid_function)
plt.scatter(df['x'], df['y'], c=df['y'], cmap='rainbow', edgecolors='b')
```