Лабораторна робота 1.3: Логістична регресія для задач класифікації.

Завдання 1.

> Оберіть один з 10 варіантів завдань відповідно до свого номеру в групі або за вказівкою викладача.

Відповідно за вказівкою викладача варіант 5.

Завдання2.

> Імпортуйте необхідні бібліотеки

> Завантажте дані відповідно до вашого варіанту

Імпортовуємо необхідні бібліотеки та заватажуємо дані класифікації електроних листів на спам та не-спам.

> Візуалізуйте дані для розуміння їх структури

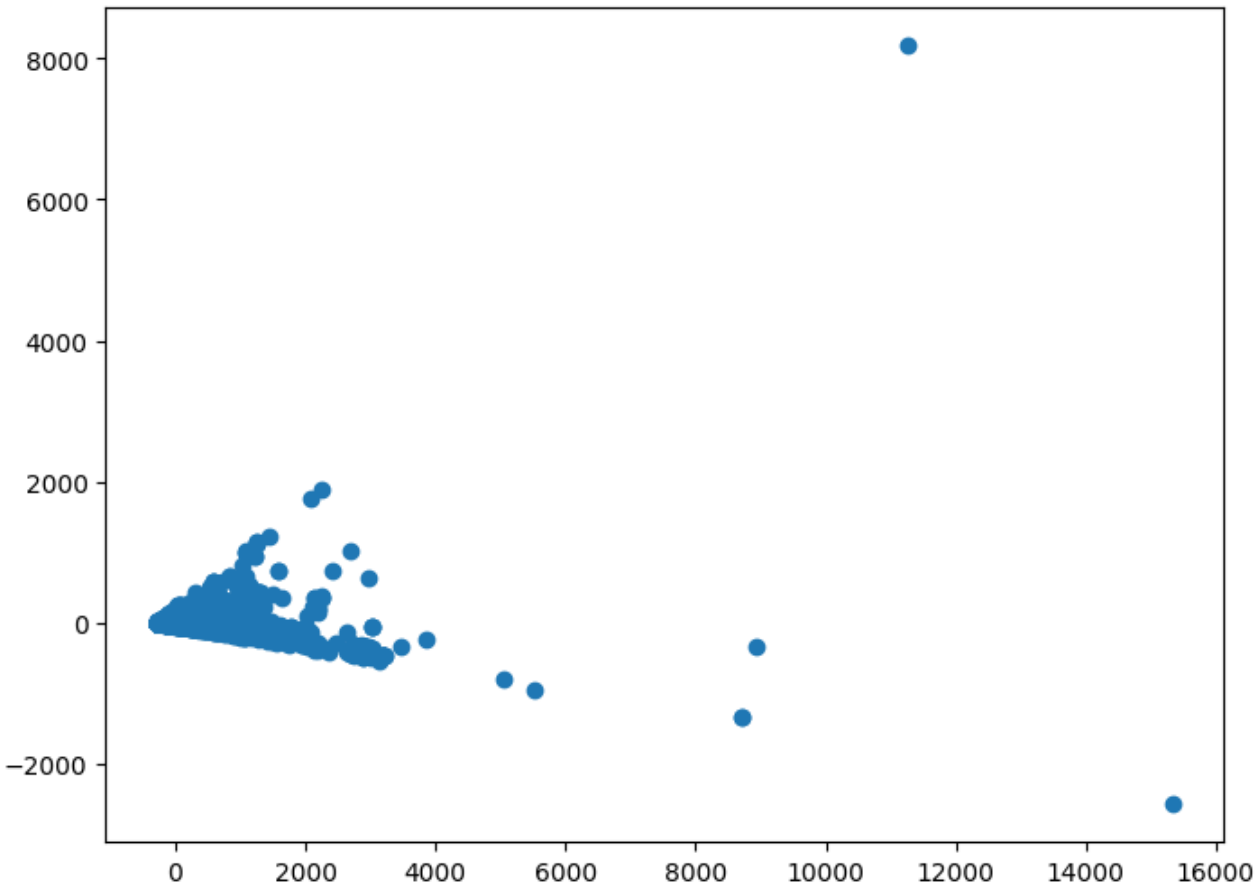

Рисунок 1 — Візуалізація даних після PCA обробки

> Поділіть дані на тренувальну та тестову вибірки

Дані розділено за допомогою train_test_split модулю sklearn.module_selection.

Завдання 3.

> Реалізуйте сигмоїдну функцію

Функцію реалізовано за формулою:

$$g(z) = \frac{1}{1+e^{-z}}$$

Сигмоїдна функція перетворює будь-яке вхідне значення в число між 0 і 1, що можна інтерпретувати як імовірність належності до класу 1.

> Реалізуйте функцію обчислення вихідних значень моделі

Функція реалізована за формулою:

$$f_{w,b}(x) = g(w \cdot x + b)$$

> Реалізуйте функцію обчислення вартості

Для логістичної регресії функція вартості обчислюється за формулою:

$$J(w,b) = -\frac{1}{m}\sum_{i=0}^{m-1}[y^{(i)}\log(f_{w,b}(x^{(i)})) + (1-y^{(i)})\log(1-f_{w,b}(x^{(i)}))]$$

Ця функція має такі властивості:

- Якщо $\mathrm{y}^{(i)} = 1$ і $\mathrm{f}_{w,b}(x^{(i)})$ близьке до 1, втрати близькі до 0
- Якщо $\mathrm{y}^{(i)} = 1$ і $\mathrm{f}_{w,b}(x^{(i)})$близьке до 0, втрати прямують до нескінченності
- Якщо $\mathrm{y}^{(i)} = 0$ і $\mathrm{f}_{w,b}(x^{(i)})$ близьке до 0, втрати близькі до 0
- Якщо $\mathrm{y}^{(i)} = 0$ і $\mathrm{f}_{w,b}(x^{(i)})$ близьке до 1, втрати прямують до нескінченності

> Реалізуйте функцію обчислення градієнту

Реалізовано за формулами:

$$\frac{\partial J(w,b)}{\partial w_j} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})x_j^{(i)}$$

$$\frac{\partial J(w,b)}{\partial b} = \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})$$

> Реалізуйте функцію градієнтного спуску

Реалізовано за формулами:

$$w_j = w_j - \alpha\frac{\partial J(w,b)}{\partial w_j}$$

$$b = b - \alpha\frac{\partial J(w,b)}{\partial b}$$

Завдання 4.

> Використайте ipywidgets для створення інтерфейсу з наступними елементами:

>> Вибір параметрів навчання (швидкість навчання, кількість ітерацій)

За допомогою повзунків можна обрати необхідні параметри навчання. Старт процесу навчання за визначеними параметрами виконується по натисканню клавіші «Навчити модель». Швидкість навчання моделі задає параметр alpha, від 0.0001 до 1 з кроком 0.0001. Ітерації задаються від 100 до 10000, крок 100.

>> Візуалізація процесу навчання

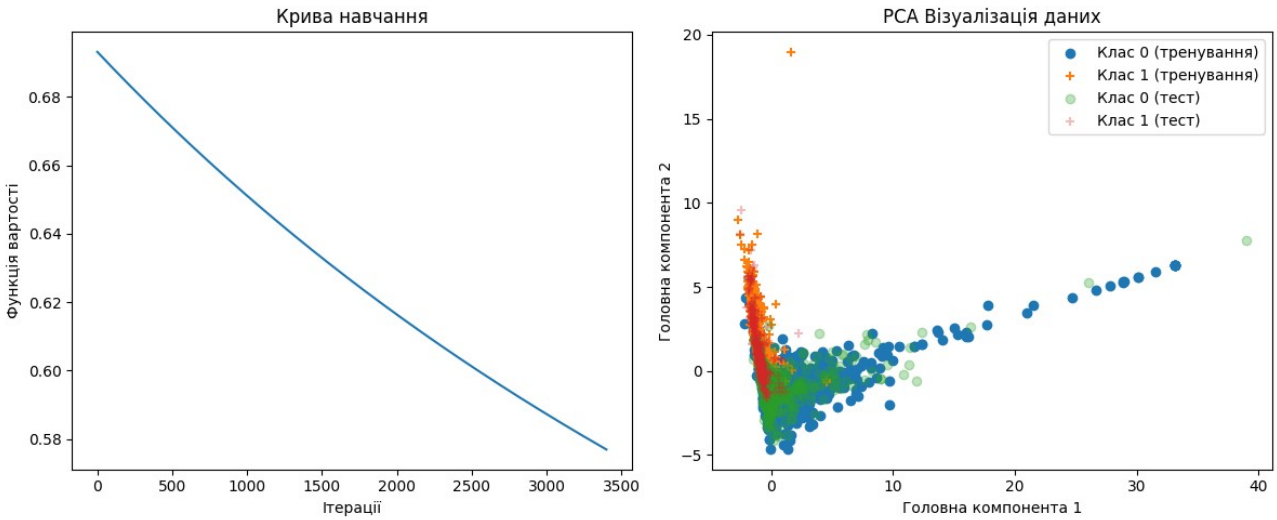

Рисунок 2 — Візуалізація кривої навчання та PCA даних

>> Вибір параметру регуляризації

Параметр регуляризації (lambda) можна обрати за допомогою повзунка. Мінімальне значення: 0; максимальне: 10; крок: 0.1.

>> Візуалізація межі прийняття рішень

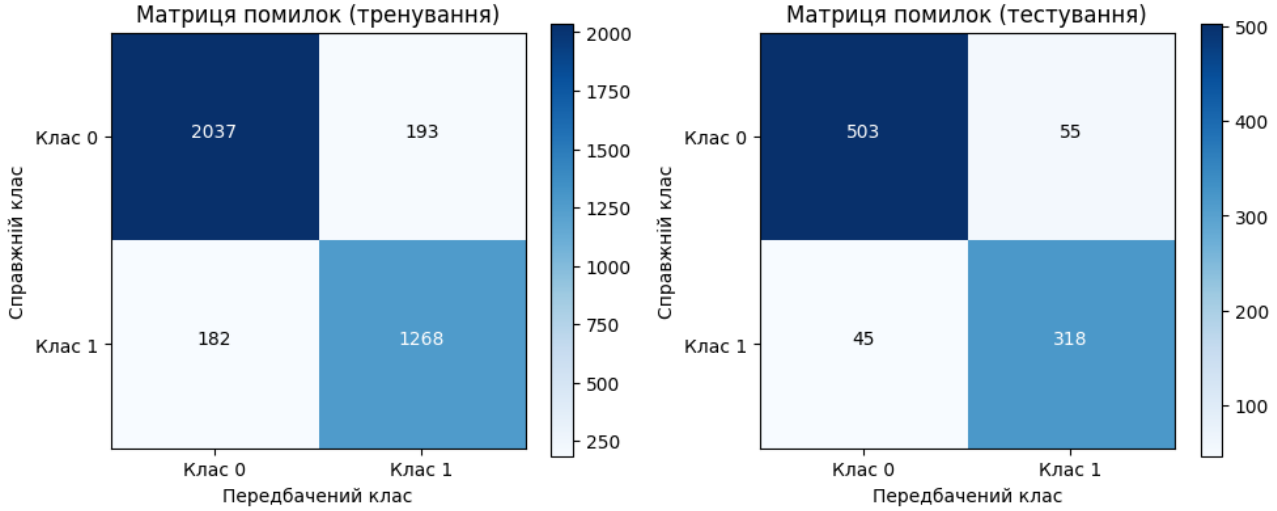

Рисунок 3 — Візуалізація матриці помилок

Завдання 5.

> Ініціалізуйте початкові значення параметрів

Початкові значення параметрів ініціалізуються при виконанні завдання 4, за допомогою повзунків.

> Виконайте алгоритм градієнтного спуску для знаходження оптимальних параметрів

Навчання моделі відбувалось за параметрів: alpha=0.01, iterations=1000, lambda=0.0

Iteration 0: Cost 0.693101

Iteration 100: Cost 0.688511

Iteration 200: Cost 0.684013

Iteration 300: Cost 0.679606
Iteration 400: Cost 0.675285
Iteration 500: Cost 0.671050
Iteration 600: Cost 0.666899
Iteration 700: Cost 0.662828
Iteration 800: Cost 0.658836
Iteration 900: Cost 0.654921

> Обчисліть точність моделі на тренувальній та тестовій вибірках
Точність на тренувальній вибірці: 89.81%
Точність на тестовій вибірці: 89.14%

> Визначте інші метрики якості (precision, recall, F1-score)
Тренувальні метрики: Precision=0.8679, Recall=0.8745, F1-score=0.8712
Тестові метрики: Precision=0.8525, Recall=0.8760, F1-score=0.8641

> Проаналізуйте вплив параметра регуляризації на результати
У цьому експерименті регуляризація не мала помітного впливу на результати, оскільки не виникало перенавчання моделі або основний вплив мали швидкість навчання та кількість ітерацій.

Завдання 6.
> Візуалізуйте межу прийняття рішень на тренувальних та тестових даних

> Проаналізуйте криву навчання (learning curve)
Модель поступово навчається і зменшує помилку. Проте досягнутий рівень помилки (0.58) може свідчити про складність задачі або недостатню складність моделі. Є потенціал для покращення якості моделі.

> Зробіть висновки щодо якості моделі та запропонуйте шляхи її вдосконалення
Навчання моделі з alpha=0.01, iterations=3400, lambda=0.0
Функція вартості поступово знижується, що свідчить про навчання моделі без переобчислення.
Точнічть моделі склала 90% для тренувальної та 89.14% для тестової, що говорить про хорошу здатність до узагальнення.
Метрика F1 0.864 говорить про збалансовану точність, а значення precision та recall без перекосу та високі.

> Чим відрізняється логістична регресія від лінійної регресії?
Лінійна регресія використовується для задач прогнозування неперервних значень, а логістична регресія — для класифікації.

> У чому полягає роль сигмоїдної функції в логістичній регресії?
Функція перетворює будь-яке значення числа у значення діапазону 0 або 1 для подальшої інетрпритації як ймовірність належності до позитивного класу.

> Чому для логістичної регресії не використовується функція вартості, як у лінійній регресії?

У лінійній регресії використовують сенредньоквадратичну помилку, але це не підходить для логістичної регресії через сигмоїдну функції. Через неї градієнт буде пласким та навчання моделі буде поганим.

> Які є способи запобігання перенавчанню (overfitting) в логістичній регресії?

Зменшення кількості ознак, збільшення кількості даних, рання зупинка навчання.

> Як впливає параметр регуляризації λ на модель?

В залежності від значення спрощує моделі, може зменшити ймовірність перенавчання.

> Як обчислити точність, precision, recall та F1-score для моделі класифікації?

Ці значення обчислюються на онові матриці помилок:

Accuracy: (TP + TN)/(TP + TN + FP + FN);

Precision: TP / (TP+FP);

Recall: TP / (TP + FN);

F1-score: 2 · (Precision · Recall) / (Precision + Recall),

де TP — істинно позитивні, FP — хибно позитивні, FN — хибно негативні, TN — істинно негативні.

> Що таке матриця помилок (confusion matrix) і як її інтерпретувати?

Це табличка 2x2, що показує наскільки модель правильно передбачила позитивно (TP), помилково передбачила позитивно (FP), помилково негативно (FN) та правильно негативно (TN).

> Які переваги та недоліки логістичної регресії порівняно з іншими методами класифікації?

З переваг це простота реалізації, швидість тренування. З недоліків: добре тільки при лінійній розіленості, не така ефективна для нелінійних залежностей.

> У яких випадках доцільно використовувати логістичну регресію, а в яких інші методи класифікації?

Доцільно, якщо потрібна інтерпретованість моделі та дані чисті, без складних структур.

Інші методи доцільно, коли важливіша висока точність та присутня складна взаємодія між ознаками.

Висновки:

В ході виконання було реалізовано логістичну регресію для виявлення спаму в електроних листах. Дані попередньо було оброблено методом PCA для зменшення розмірності та візуалізації. Також було поетапно реалізовано сигмоїдну функцію, функцію вартості, градієнтний спуск та інтерфкйс користувача для задання початкових параметрів

навчання моделі. У процесі було помітно гарний хід навчання моделі: функція втрат поступово зменшувалась, а модель досягала точності вище 90% на тренувальних вправах та вище 89% на тестових. Ключві метрики Precision, Recall та F1-score показали хороші значення. Параметр регуляризації мав незначний вплив на модель, що може свідчити про відсутність перенавченості моделі.