**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

Кафедра математичної інформатики

**Звіт**

до лабораторної роботи № 2

Виконав студент 1-го курсу

ОНП “Штучний інтелект”

Ходаков Максим Олегович

Київ – 2025

# Звіт по лабораторній роботі

**Тема:** Логістична регресія на датасеті Iris з використанням L1 та L2 регуляризації

## 1. Постановка задачі

Метою роботи є побудова моделі **логістичної регресії** для багатокласової класифікації квітів Iris за їх морфологічними ознаками. Завдання включає:

* завантаження та підготовку датасету;
* побудову моделей логістичної регресії з L1- та L2-регуляризацією;
* навчання та оцінювання якості моделей за допомогою метрик класифікації (accuracy, precision, recall, F1-score);
* візуалізацію результатів (матриці плутанини, криві навчання та валідації);
* аналіз можливих проблем перенавчання (overfitting) та недонавчання (underfitting).

## 2. Опис датасету

Використано відомий датасет [Iris dataset (Fisher, 1936)](https://www.kaggle.com/datasets/uciml/iris).  
У наборі даних 150 спостережень трьох видів квітів:

* Iris-setosa,
* Iris-versicolor,
* Iris-virginica.

Кожна квітка описується чотирма числовими ознаками:

* **SepalLengthCm** — довжина чашолистка (см),
* **SepalWidthCm** — ширина чашолистка (см),
* **PetalLengthCm** — довжина пелюстки (см),
* **PetalWidthCm** — ширина пелюстки (см).

Цільова змінна (**Species**) — категоріальна (3 класи).

## 3. Теоретичні відомості

### Логістична регресія

Логістична регресія — статистичний метод, що моделює ймовірність належності об’єкта до певного класу.  
Для двокласової задачі використовується **логістична функція** (сигмоїда):

P(y=1|x) = 1 / (1 + exp(-(w·x + b)))

У випадку багатокласової класифікації застосовується **softmax-функція**, яка повертає ймовірності для всіх класів.

### Регуляризація

Регуляризація контролює величину ваг моделі, запобігаючи перенавчанню.

* **L2 (Ridge)** — штрафує суму квадратів ваг:  
  Loss = LogLoss + λ \* ||w||²  
  Зменшує всі коефіцієнти плавно.
* **L1 (Lasso)** — штрафує суму абсолютних значень ваг:  
  Loss = LogLoss + λ \* ||w||₁  
  Може занулювати окремі коефіцієнти → виконує відбір ознак.

Параметр **C** в sklearn — це зворотна сила регуляризації:

* малий C → сильна регуляризація;
* великий C → слабка регуляризація.

## 4. Використані технології

* **Python 3.9+**
* Бібліотеки:
  + pandas, numpy — для роботи з даними;
  + scikit-learn — моделі, метрики, розбиття даних;
  + matplotlib, seaborn — візуалізація.

## 5. Хід виконання роботи

1. **Завантаження даних** — датасет Iris у форматі CSV. Колонка Id видалена.
2. **Кодування міток** — рядкові назви видів квітів перетворені у числа (0, 1, 2).
3. **Поділ даних** — 60% train, 20% validation, 20% test (з урахуванням пропорцій класів).
4. **Моделі**:
   * Logistic Regression з L2 (solver = lbfgs);
   * Logistic Regression з L1 (solver = liblinear).
5. **Навчання та оцінка**:
   * побудовані classification\_report для train/val/test;
   * побудовані confusion matrix для тесту;
   * перевірено можливе перенавчання через різницю F1 на train і validation.
6. **Візуалізація**:
   * крива навчання (Learning curve): як змінюється якість при зростанні обсягу train;
   * валідаційна крива (Validation curve): залежність F1 від параметра C.

## 6. Результати

* **L2-модель** показала стабільні результати на train/val/test з високим F1 (~0.95–1.0).
* **L1-модель** теж працює добре, але має тенденцію зануляти деякі коефіцієнти (ознаки), що корисно для інтерпретації.
* На кривих навчання видно, що з ростом кількості прикладів розрив між Train і Validation F1 зменшується, отже модель добре узагальнює.
* На кривій валідації видно оптимальну зону C (приблизно від 1 до 10), де модель не перенавчається і не недонавчається.
* **Матриця плутанини** показала, що модель іноді плутає versicolor та virginica (класична проблема Iris).

## 7. Висновки

1. Логістична регресія ефективно класифікує дані Iris, досягаючи високої точності (>95%).
2. L2-регуляризація надає стабільніші результати, тоді як L1 виконує відбір ознак.
3. Використання **валідаційної вибірки** дозволяє відстежувати overfitting: суттєвої різниці між train і val F1 не виявлено.
4. Побудова кривих навчання й валідації дозволила виявити оптимальне значення параметра C та підтвердити, що модель не страждає від недонавчання.
5. Отримані результати можуть бути основою для подальшого використання логістичної регресії на схожих багатокласових задачах.
6. Код програми: <https://github.com/maksymkhodakov/MLLab2>