Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни

«Прикладні задачі машинного навчання»

«Класифікація методом k найближчих сусідів

і набір даних Digits, частина 1»

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Нестерук Андрій Олександрович

Київ 2023

**Лабораторна робота 4**

**Класифікація методом k найближчих сусідів**

**і набір даних Digits, частина 1**

**Постановка завдання.**

1. Для дослідження даних, візуалізувати їх. Вивести зображення перших 24 і 36 цифр з набору;
2. Розбити дані на навчальні та тестові, за замовчуванням train\_test\_split резервує 75% даних для навчання і 25% для тестування, змінити це;
3. Створити та навчити модель;
4. Виконати прогнозування класів;
5. Порівняти прогнозовані цифри з очікуваними для перших 20, 24, 36 тестових зразків;
6. Пояснити результат, застосувати метрики точності моделі:
   1. Метод score оцінювача;
   2. Матриця невідповідностей;
7. Вивести звіт класифікації;
8. Використати декілька моделей KNeighborsClassifier, SVC і GaussianNB для пошуку найкращої;
9. Налаштувати гіперпараметр K в KNeighborsClassifier.

**Хід роботи.**

1. Для дослідження даних, візуалізую їх. Виводжу зображення перших 36 цифр з набору;

Для початку виводжу двовимірний масив зображення 36 цифри з набору. Потім за допомогою метою imshow() візуалізую зображення, задане в масиві для перших 36 цифр. Отримуємо такі результати:

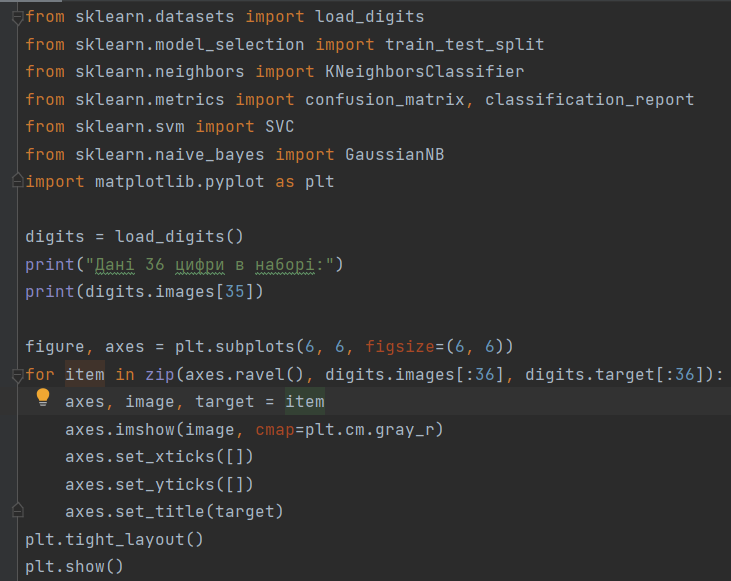


Рисунок 1. Код візуалізації перших 36 цифр з набору.

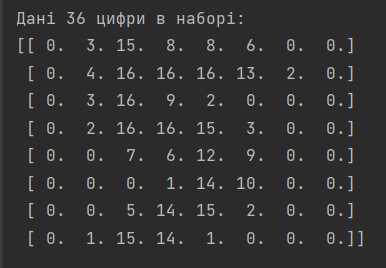


Рисунок 2. Масив зображення 36-ої цифри в наборі.

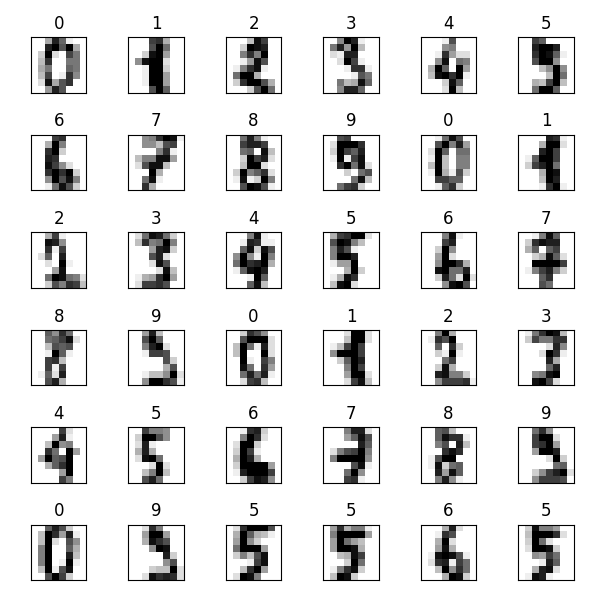


Рисунок 3. Візуалізація перших 36 цифр з набору.

Бачимо, що спочатку у нас йдуть тричі послідовність цифр 0-9, а потім цифри починають з’являтися упереміш.

1. Розбиваю дані на навчальні та тестові, за замовчуванням train\_test\_split резервує 75% даних для навчання і 25% для тестування, змінюю це;

Розіб’ю дані на навчальні й тестові за замовчуванням, щоб переконатися у відсотковому співвідношенні, а також зміню параметр цього відношення, щоб тестова вибірка отримувала 20% даних.

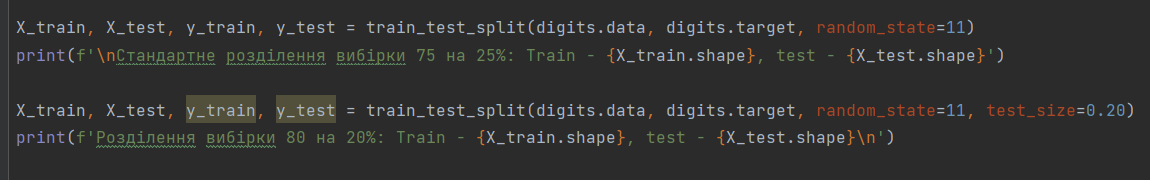


Рисунок 4. Стандартне розділення даних та розділення 80% на 20%.

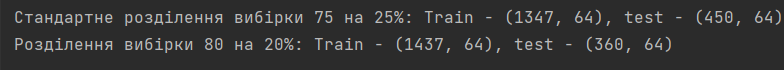


Рисунок 5. Результат розділення даних на навчальні й тестові.

Отримуємо для стандартного розділення 1347 навчальних даних і 450 тестових. Маємо відсоток навчальних даних,

Отже, переконалися, що стандартне розділення має 75% на 25%. Для розділення 80 на 20% порахуємо відсоток тестових даних:

Тобто дані дійсно розділилися так, як було вказано.

1. Створити та навчити модель;

Створюю модель KNeighborsClassifier для алгоритму К найближчих сусідів. Далі треную модель за допомогою попередньо розділених даних, відповідно завантажую навчальні дані.

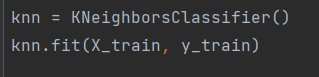


Рисунок 6. Створення моделі KNeighborsClassifier.

1. Виконати прогнозування класів;

Для прогнозування класів використовую метод predict() і тестові дані.



Рисунок 7. Прогнозування класів.

1. Порівняти прогнозовані цифри з очікуваними для перших 20, 24, 36 тестових зразків;

Виводимо список перших 36 передбачених значень та очікуваних.

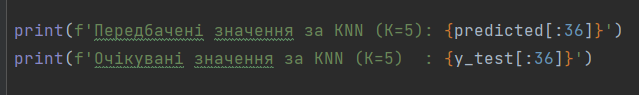


Рисунок 8. Виведення передбачених та очікуваних значень.



Рисунок 9. Результат виведення передбачених та очікуваних значень.

Бачимо, що передбачені значення від очікуваних у перших 36 зразках відрізняються лише в одному випадку. Оцінювач передбачив 5, а насправді була цифра 3. Підвищену складність такої оцінки можна пояснити тим, що 5 і 3 доволі схожі цифри, і за відповідного стилю письма дійсно може бути важче розрізнити такі цифри.

1. Пояснити результат, застосувати метрики точності моделі:
   1. Метод score оцінювача;

Застосуємо метод score.

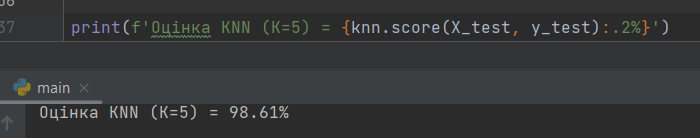


Рисунок 10. Застосування методу score.

Бачимо, що ефективність доволі висока, що ми вже побачили пунктом вище в перших 36 зразках.

* 1. Матриця невідповідностей;

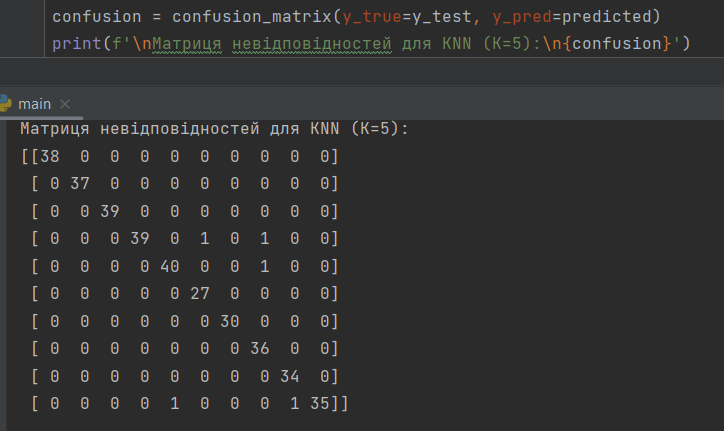


Рисунок 11. Застосування матриці невідповідностей.

За визначенням матриця невідповідностей є такою, що дорівнює кількості спостережень, про які відомо, що вони належать до групи й передбачені у групі . Бачимо, що абсолютна більшість передбачені правильно, адже правильні результати знаходяться на головній діагоналі. Хоча є поодинокі невідповідності, в чому ми переконуємося з двох попередніх пунктів.

1. Виводжу звіт класифікації;

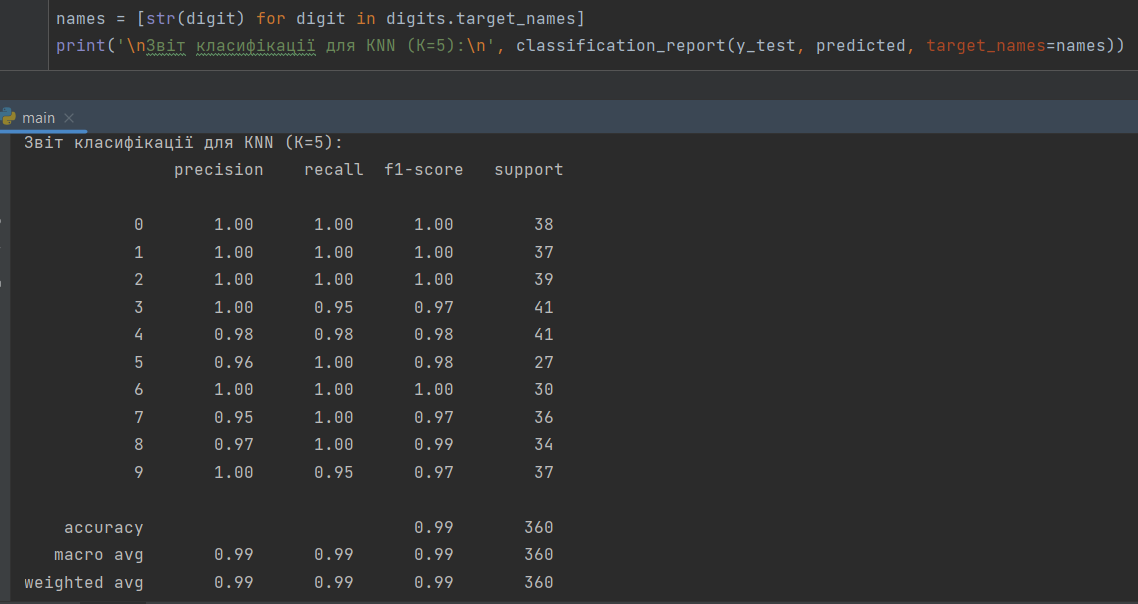


Рисунок 12. Застосування звіту класифікації.

Бачимо, що precision (серед усього, що було передбачено як позитивне, підраховує відсоток, який виявився правильним) найгірша у цифри 7, трошки краща 5, далі 8 і 4, а решта розпізнаних цифр завжди виявлялися саме тими; recall (рахує, скільки з того, що насправді є позитивним, вдалося віднайти моделі) найгірший для 9 і 3, трошки кращий для 4, для решти цифр вдалося віднайти усі зразки; відповідно бачимо f1-score, що є середнім гармонійним для precision та recall, має найвищі значення для 0, 1, 2 і 6, адже для цих цифр усі зразки були розпізнані і розпізнані правильно. Бачимо, що середні значення для усіх величин є 0.99, що є досить високим показником, тобто модель працює ефективно.

1. Використовую декілька моделей (KNeighborsClassifier, SVC і GaussianNB) для пошуку найкращої. Оцінку KNeighborsClassifier вже маємо, тож застосуємо SVC й GaussianNB.

Тож створюю відповідні моделі, роблю передбачення. Для оцінки їх ефективности я обрав метод score й звіт класифікації.

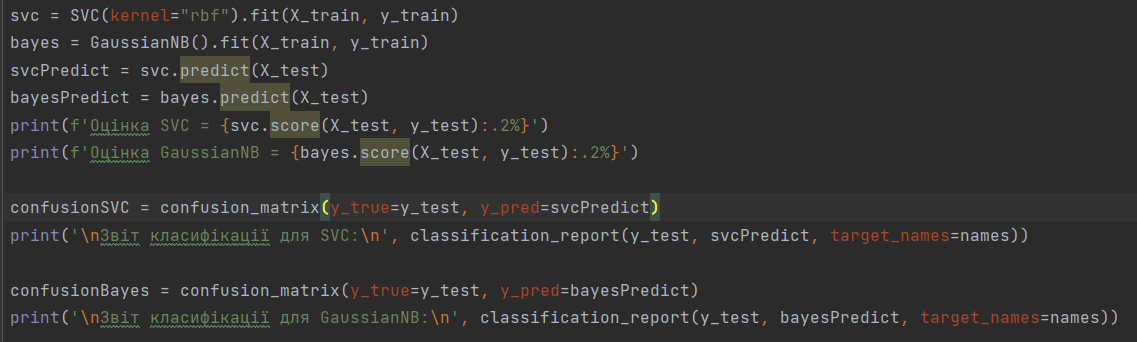


Рисунок 13. Застосування моделей SVC і GaussianNB.

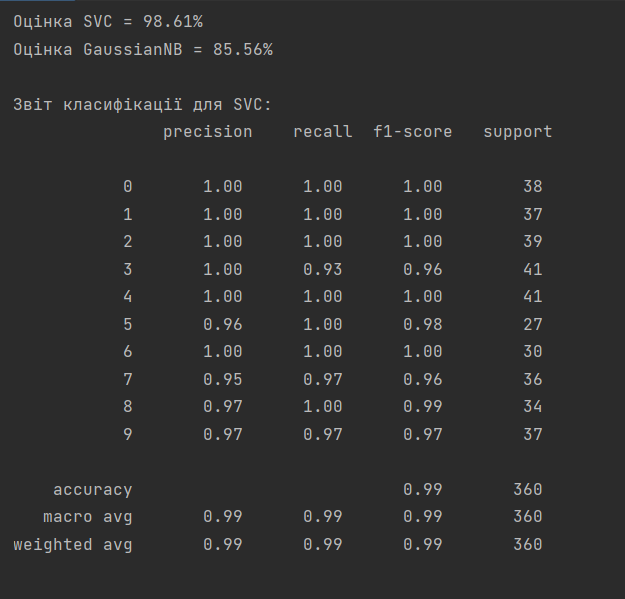


Рисунок 14. Оцінки score й звіт класифікації для SVC.

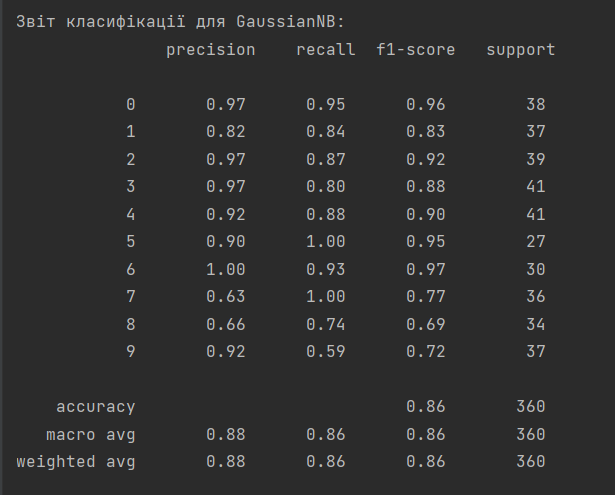


Рисунок 15. Звіт класифікації для GaussianNB.

Бачимо, що SVC працює так само ефективно, як і KNN (K=5) з оцінкою 98.61%. У той час як GaussianNB працює доволі неефективно з оцінкою 85.56%. За звітом класифікації що в KNN, що в SVC precision не 1 в чотирьох цифрах, а recall – у трьох. Причому спільні проблеми з розпізнаванням за precision в них за цифрами 5, 7, 8; за recall – 3 й 9. У SVC нема проблеми з розпізнаванням цифри 4, як у KNN.

Тож для даного набору поки що найкращими моделями є KNN (K=5) й SVC.

1. Налаштувати гіперпараметр K в KNeighborsClassifier.

За замовчуванням гіперпараметр K у нас був 5. За допомогою експерименту визначимо, який параметр буде найкращий у цьому наборі. Я буду досліджувати K від 2 до 10 включно.

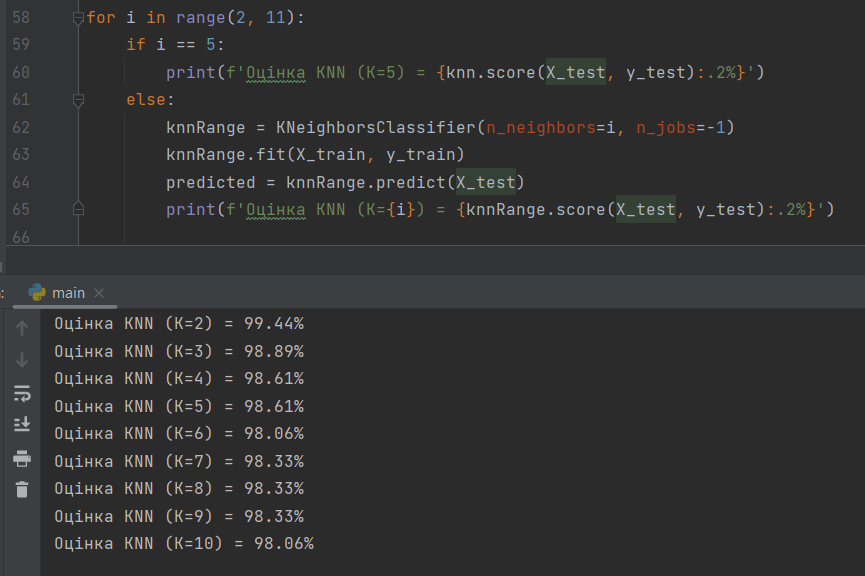


Рисунок 16. Дослідження гіперпараметра K.

У результаті дослідження виявили, що найкращим для даного набору виявився KNN з K=2. Методом score оцінка його ефективности 99.44%. Це краще за KNN (K=5) та SVC.

**Висновок.**

Отже, у цій роботі я навчився реалізовувати основні етапи машинного навчання: вибір даних для навчання моделі, завантаження та аналіз даних, розбиття даних для навчання і тестування, вибір, побудова й навчання моделі, формування прогнозів - проводити оцінку результатів, опрацьовувати кілька класифікаційних моделей для вибору найкращої.

У результаті лабораторної роботи було вивчено роботу з модулем scikit-learn, візуалізовано дані для досліджуваного набору розпізнавання цифр, проведено навчання й тестування за допомогою класифікаторів K-NN, SCV, GaussianNB. Також був досліджений параметр K для KNN. У результаті дослідження отримали, що KNN з K=2 є найкращим оцінювачем для даних. Використовуючи програмний засіб PyCharm, мову програмування Python з модулями, отримуємо коректний результат.