МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №8

з дисципліни Спеціалізовані мови програмування

на тему

Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python

Виконав:

студент групи РІ-21сп

Андрій ПІГУЛЯК

Львів – 2024

**Мета:** Розробка додатка для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм)

**Хід роботи**

1. Обрати CSV-набір даних для візуалізації, який містить відповідну інформацію для створення змістовних графіків та діаграм.

2. Завантажити дані з CSV-файлу у програму Python, використовуючи бібліотеки, такі як Pandas, для обробки.

3. Провести аналіз даних, визначивши екстремальні значення у стовпцях.

4. Обрати типи візуалізацій, які найкраще підходять для представлення даних, такі як лінійні графіки, гістограми, стовпчикові та секторні діаграми.

5. Підготувати дані для візуалізації: виконати фільтрацію, виправлення помилок або трансформацію, якщо необхідно.

6. Побудувати базову візуалізацію з використанням Matplotlib, наприклад, лінійний графік чи гістограму, для перевірки коректності відображення даних.

7. Створити розширені візуалізації, експериментуючи з налаштуваннями графіків і додатковими функціями Matplotlib.

8. Навчитися створювати кілька піддіаграм в одному малюнку, щоб порівняти різні візуалізації поруч.

9. Реалізувати експорт візуалізацій як зображень (PNG, SVG) або інтерактивних веб-додатків (HTML) для подальшого використання чи обміну.

1. **Опис і структура програми:**

**Структура директорій і файлів**

1. **chart\_builder/**
   * \_\_init\_\_.py: Порожній файл, що позначає директорію як Python-пакет.
   * base.py: Абстрактний базовий клас для створення діаграм.
   * correlation.py: Реалізація побудови кореляційної матриці.
   * histogram.py: Реалізація побудови гістограми.
   * multiple.py: Реалізація побудови кількох піддіаграм.
   * scatter.py: Реалізація побудови діаграми розсіювання.
2. **Кореневий каталог:**
   * chart\_saver.py: Клас для збереження побудованих діаграм у різних форматах.
   * data\_handler.py: Завантаження даних із CSV-файлів.
   * main.py: Головний файл для запуску програми.
   * user\_input.py: Клас для роботи з введенням користувача.
   * wine\_visualizer.py: Основний модуль для візуалізації даних.
3. **Каталог даних:**
   * csv/winequality-red.csv
   * csv/winequality-white.csv
     1. **chart\_builder/base.py**

* Містить абстрактний клас ChartBuilder із методом build\_chart(), який визначає інтерфейс для створення діаграм.
  + 1. **chart\_builder/correlation.py**
* Реалізує клас CorrelationMatrixBuilder для побудови кореляційної матриці. Використовуються matplotlib і seaborn для створення теплокарти кореляційних коефіцієнтів.
  + 1. **chart\_builder/multiple.py**
* Реалізує клас MultipleSubplotsBuilder, який створює кілька піддіаграм: розподіл алкоголю, залежність алкоголю і якості, розподіл якості вина, та кореляційна матриця.
  + 1. **chart\_builder/scatter.py**
* Реалізує клас ScatterPlotBuilder для побудови діаграми розсіювання, що показує залежність якості вина від вмісту алкоголю.
  + 1. **chart\_saver.py**
* Надає можливість зберігати діаграми в різних форматах (png, jpg, svg, pdf, html).
  + 1. **data\_handler.py**
* Реалізує метод для завантаження даних із CSV-файлів за допомогою бібліотеки pandas.
  + 1. **user\_input.py**
* Забезпечує зручну обробку введення користувача з перевіркою коректності введених даних.
  + 1. **wine\_visualizer.py**
* Основний модуль програми, що дозволяє користувачеві вибирати тип вина (червоне чи біле) та будувати бажану діаграму.

1. **Функціонал:**
2. **Завантаження даних:**   
   Програма завантажує дані про якість червоного або білого вина з відповідного CSV-файлу.
3. **Вибір типу діаграми:**   
   Користувач може побудувати одну з наступних діаграм:

* Гістограма розподілу якості вина.
* Діаграма розсіювання залежності алкоголю та якості.
* Кореляційна матриця.
* Кілька піддіаграм.

1. **Візуалізація:**   
   Використовуються бібліотеки matplotlib і seaborn для побудови діаграм із налаштуванням розміру, кольорів і стилів.
2. **Збереження результатів:**   
   Користувач може зберегти створену діаграму у вибраному форматі, наприклад, png, svg, html.

**3. Код програми**

1. base.py

from abc import ABC, abstractmethod  
  
class ChartBuilder(ABC):  
 @abstractmethod  
 def build\_chart(self, data):  
 pass

1. **histogram.py**

class HistogramBuilder(ChartBuilder):  
 def build\_chart(self, data):  
 fig = plt.figure(figsize=(8, 6))  
 plt.hist(data['quality'], bins=range(3, 10), edgecolor='black', color='skyblue')  
 plt.title("Розподіл якості вин")  
 plt.xlabel("Якість вина")  
 plt.ylabel("Кількість")  
 plt.xticks(range(3, 10))  
 return fig

1. **user\_input.py**Фрагмент обробки користувацького вводу:

import pandas as pd  
  
class DataHandler:  
 @staticmethod  
 def load\_data(file\_path):  
 return pd.read\_csv(file\_path, sep=';')

**4.** **Приклад роботи:**

**Приклад 1:** Вибираємо тип CSV



Рис. 1 Обираємо червоний тап вина

**Приклад 2:** Відображення кількох піддіаграм

A screenshot of a computer

Description automatically generated

Рис. 2 вибираємо «кілька піддіаграм»

**Приклад 3:** Результат зображення кількох піддіаграм

A collage of graphs and charts

Description automatically generated

Рис. 3 Кілька піддіаграм

**Висновок:** У ході виконання лабораторної роботи було реалізовано модульну програму для аналізу та візуалізації даних. Програма дозволяє будувати діаграми різного типу з подальшим їх збереженням. Застосування шаблону проектування "Абстрактна фабрика" забезпечило легке розширення функціональності програми шляхом додавання нових типів діаграм.