# **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА” ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ** **Кафедра ІСМ**

# **Звіт**

# **до лабораторної роботи №8**

# **З дисципліни “Спеціальні мови програмування”**

**Виконав:  
ст.гр.ІТ-31  
Шельвах Максим  
  
Прийняв:  
Щербак С.С.**

# **Львів — 2023**

**Тема роботи:** Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python.

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

## План роботи

Завдання 1: Вибір CSV-набору даних

Оберіть CSV-набір даних, який ви хочете візуалізувати. Переконайтеся, що він містить відповідні дані для створення змістовних візуалізацій.

Завдання 2: Завантаження даних з CSV

Напишіть код для завантаження даних з CSV-файлу в ваш додаток Python. Використовуйте бібліотеки, такі як Pandas, для спрощення обробки даних.

Завдання 3: Дослідження даних

Визначте екстремальні значення по стовцям

Завдання 4: Вибір типів візуалізацій

Визначте, які типи візуалізацій підходять для представлення вибраних наборів даних. Зазвичай це може бути лінійні графіки, стовпчикові діаграми, діаграми розсіювання, гістограми та секторні діаграми.

Завдання 5: Підготовка даних

Попередньо обробіть набір даних за необхідністю для візуалізації. Це може включати виправлення даних, фільтрацію, агрегацію або трансформацію.

Завдання 6: Базова візуалізація

Створіть базову візуалізацію набору даних, щоб переконатися, що ви можете відображати дані правильно за допомогою Matplotlib. Розпочніть з простої діаграми для візуалізації однієї змінної.

Завдання 7: Розширені візуалізації

Реалізуйте більш складні візуалізації, виходячи з характеристик набору. Поекспериментуйте з різними функціями Matplotlib та налаштуваннями.

Завдання 8: Декілька піддіаграм

Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

Завдання 9: Експорт і обмін

Реалізуйте функціональність для експорту візуалізацій як зображень (наприклад, PNG, SVG) або інтерактивних веб-додатків (наприклад, HTML).

Хід роботи

**Реалізований код:**

User\_interface.py

# Імпортуємо класи та модулі з інших файлів для взаємодії з різними аспектами аналізу даних.

from .methods.data\_loader import DataLoader

from .methods.data\_visualizations import VisualizationSelector

from .methods.data\_preprocessor import DataPreprocessor

from .methods.data\_explorer import DataExplorer

from .methods.base\_visualization import BasicVisualizer

from .methods.advanced\_visualization import AdvancedVisualizer

from .methods.subplots import MultipleSubplots

from .methods.export import Exporter

class DataAnalysisMenu:

def \_\_init\_\_(self):

# Ініціалізуємо об'єкт для завантаження даних.

self.loader = DataLoader()

# Ініціалізуємо об'єкт для обробки даних (поки не обрано конкретну операцію обробки).

self.preprocessor = None

def display\_menu(self):

# Відображення головного меню програми.

menu\_options = {

1: "Explore Data",

2: "Choose Visualizations",

3: "Data Preprocessing",

4: "Basic Visualization",

5: "Advanced Visualization",

6: "Create Subplots",

7: "Export Visualization",

8: "Exit"

}

print("Menu:")

for num, option in menu\_options.items():

print(f"{num}. {option}")

def explore\_data(self):

# Запуск режиму дослідження даних.

explorer = DataExplorer(self.loader.data)

explorer.explore\_data()

def choose\_visualizations(self):

# Запуск режиму вибору візуалізацій.

visual\_selector = VisualizationSelector(self.loader.data)

visual\_selector.choose\_visualizations()

def preprocess\_data(self):

# Запуск режиму попередньої обробки даних.

self.preprocessor = DataPreprocessor(self.loader)

self.preprocessor.preprocess()

def basic\_visualization(self):

# Запуск режиму базової візуалізації (після попередньої обробки даних).

if self.preprocessor:

basic\_visualizer = BasicVisualizer(self.preprocessor.processed\_data)

basic\_visualizer.visualize()

else:

print("Data must be preprocessed first. Choose option 3 for Data Preprocessing.")

def advanced\_visualization(self):

# Запуск режиму розширеної візуалізації (після попередньої обробки даних).

if self.preprocessor:

advanced\_visualizer = AdvancedVisualizer(self.preprocessor.processed\_data)

advanced\_visualizer.advanced\_visualizations()

else:

print("Data must be preprocessed first. Choose option 3 for Data Preprocessing.")

def create\_subplots(self):

# Запуск режиму створення піддіаграм (після попередньої обробки даних).

if self.preprocessor:

subplots\_creator = MultipleSubplots(self.preprocessor.processed\_data)

subplots\_creator.create\_subplots()

else:

print("Data must be preprocessed first. Choose option 3 for Data Preprocessing.")

def export\_visualization(self):

# Запуск режиму експорту візуалізації (після попередньої обробки даних).

if self.preprocessor:

exporter = Exporter(self.preprocessor.processed\_data)

exporter.export\_visualization(file\_format="png")

else:

print("Data must be preprocessed first. Choose option 3 for Data Preprocessing.")

def run\_menu(self):

# Головний цикл програми для взаємодії з користувачем.

while True:

self.display\_menu()

choice = input("Enter the number of your choice: ")

if choice.isdigit():

choice = int(choice)

if choice == 1:

self.explore\_data()

elif choice == 2:

self.choose\_visualizations()

elif choice == 3:

self.preprocess\_data()

elif choice == 4:

self.basic\_visualization()

elif choice == 5:

self.advanced\_visualization()

elif choice == 6:

self.create\_subplots()

elif choice == 7:

self.export\_visualization()

elif choice == 8:

break # Вихід з програми

else:

print("Invalid choice. Please enter a valid number.")

else:

print("Invalid input. Please enter a number.")

main.py

from .interface.user\_interface import DataAnalysisMenu

def main():

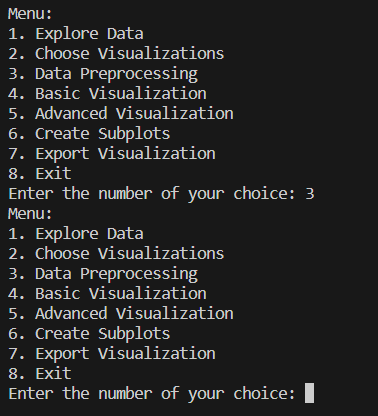
menu = DataAnalysisMenu()

menu.run\_menu()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат виконання програми:**



**Висновок:** виконавши ці завдання, я створив багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. Цей проект покращив мої навички візуалізації даних, дозволяючи досліджувати результати з різноманітними наборами даних.