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

​*Спеціалізовані мови програмування*

на тему:

“**Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python**”

Виконав студент РІ-31

**Тесліцький Ярослав**

Прийняв: Щербак С.С.

Львів – 2024

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

**Завдання лабораторної 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)

Файл запуску:

    class DataVisualizationApp:

    def \_\_init\_\_(self, file\_path):

        self.data\_loader = DataLoader(file\_path)

        self.data = self.data\_loader.get\_data()

    def preprocess\_data(self):

        self.data = self.data.dropna()

    def show\_data\_summary(self):

        self.data\_loader.explore\_data()

    def visualize\_chart(self, chart\_type, x\_column, y\_column=None):

        chart = ChartFactory.create\_chart(chart\_type, self.data, x\_column, y\_column)

        chart.plot()

        plt.show()

    def visualize\_double(self, chart\_type, x\_column1, y\_column1, x\_column2, y\_column2):

        fig, axs = plt.subplots(1, 2, figsize=(12, 5))

        chart1 = ChartFactory.create\_chart(chart\_type, self.data, x\_column1, y\_column1)

        chart2 = ChartFactory.create\_chart(chart\_type, self.data, x\_column2, y\_column2)

        plt.sca(axs[0])

        chart1.plot()

        plt.sca(axs[1])

        chart2.plot()

        plt.tight\_layout()

        plt.show()

    def export\_visualization(self, chart\_type, x\_column, y\_column, file\_name):

        chart = ChartFactory.create\_chart(chart\_type, self.data, x\_column, y\_column)

        plt.figure()

        chart.plot()

        plt.savefig(file\_name)

        plt.close()

        print(f"Chart saved as {file\_name}")

    def show\_menu(self):

        while True:

            try:

                print("\nSelect an option:")

                print("1. Show data summary and extreme values")

                print("2. Line Chart (Date vs Temperature and Humidity)")

                print("3. Bar Chart (Date vs Temperature and Humidity)")

                print("4. Scatter Chart (Date vs Temperature and Humidity)")

                print("5. Histogram (Temperature and Humidity)")

                print("6. Pie Chart (Category Column)")

                print("7. Exit")

                choice = input("Enter choice (1-7): ")

                if choice == '7':

                    print("Exiting the program.")

                    break

                elif choice == '1':

                    self.show\_data\_summary()

                elif choice in ['2', '3', '4']:

                    chart\_type = "line" if choice == '2' else "bar" if choice == '3' else "scatter"

                    self.visualize\_double(chart\_type, "date", "temperature", "date", "humidity")

                    save\_option = input("Do you want to save these charts? (y/n): ").lower()

                    if save\_option == 'y':

                        file\_name\_temp = input("Enter the file name to save the Temperature chart : ")

                        file\_name\_humidity = input("Enter the file name to save the Humidity chart : ")

                        self.export\_visualization(chart\_type, "date", "temperature", file\_name\_temp)

                        self.export\_visualization(chart\_type, "date", "humidity", file\_name\_humidity)

                elif choice == '5':

                    self.visualize\_double("histogram", "temperature", "temperature", "humidity", "humidity")

                    save\_option = input("Do you want to save these charts? (y/n): ").lower()

                    if save\_option == 'y':

                        file\_name\_temp = input("Enter the file name to save the Temperature histogram : ")

                        file\_name\_humidity = input("Enter the file name to save the Humidity histogram : ")

                        self.export\_visualization("histogram", "temperature", None, file\_name\_temp)

                        self.export\_visualization("histogram", "humidity", None, file\_name\_humidity)

                elif choice == '6':

                    self.visualize\_chart("pie", "category\_column")

                    save\_option = input("Do you want to save this chart? (y/n): ").lower()

                    if save\_option == 'y':

                        file\_name = input("Enter the file name to save the Pie chart : ")

                        self.export\_visualization("pie", "category\_column", None, file\_name)

                else:

                    print("Invalid choice. Please select a valid option.")

            except ValueError as ve:

                print(f"Invalid input: {ve}. Please try again.")

            except Exception as e:

                print(f"An error occurred: {e}. Please try again.")

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

    file\_path = "data.csv"  # Файл з даними

    app = DataVisualizationApp(file\_path)

    app.preprocess\_data()

    app.show\_menu()

**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.

GitHub: https://github.com/RomchykPrytula/SMP