Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №8

з дисципліни «Спеціалізовані мови програмування»

на тему

«Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python»

Виконав:

Ложовський В. А.

Перевірив:

Щербак С. С.

Львів 2024

**Мета:** 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)

**Реалізація:**

**Папка visualizer, файл bar\_chart.py:**

import matplotlib.pyplot as plt

from .base import Visualizer

from ..exporter import Exporter

class BarChart(Visualizer):

    def \_\_init\_\_(self, data, x\_col, y\_col):

        super().\_\_init\_\_(data)

        self.x\_col = x\_col

        self.y\_col = y\_col

    def plot(self):

        plt.bar(self.data[self.x\_col], self.data[self.y\_col])

        plt.title(f"Bar chart: {self.x\_col} vs {self.y\_col}")

        plt.xlabel(self.x\_col)

        plt.ylabel(self.y\_col)

        Exporter.export\_plot(f'output/{self.x\_col}\_vs\_{self.y\_col}\_bar\_chart.png')

        self.show()

**Папка visualizer, файл line\_chart.py:**

import matplotlib.pyplot as plt

from ..exporter import Exporter

class LineChart:

    def \_\_init\_\_(self, data, x\_col, y\_col):

        self.data = data

        self.x\_col = x\_col

        self.y\_col = y\_col

    def plot(self):

        if self.x\_col not in self.data.columns or self.y\_col not in self.data.columns:

            print(f"Columns {self.x\_col} or {self.y\_col} not found in data!")

            return

        plt.figure()

        plt.plot(self.data[self.x\_col], self.data[self.y\_col])

        plt.title(f"Line chart: {self.x\_col} vs {self.y\_col}")

        plt.xlabel(self.x\_col)

        plt.ylabel(self.y\_col)

        plt.grid(True)

        Exporter.export\_plot(f'output/{self.x\_col}\_vs\_{self.y\_col}\_line\_chart.png')

        plt.show()

**Папка visualizer, файл multi\_plot.py:**import matplotlib.pyplot as plt

from .base import Visualizer

from ..exporter import Exporter

class MultiPlotVisualizer(Visualizer):

    def \_\_init\_\_(self, data):

        super().\_\_init\_\_(data)

    def plot\_subplots(self, plots):

        fig, axes = plt.subplots(1, len(plots), figsize=(15, 5))

        for ax, (plot\_type, x\_col, y\_col) in zip(axes, plots):

            if plot\_type == 'line':

                ax.plot(self.data[x\_col], self.data[y\_col])

            elif plot\_type == 'bar':

                ax.bar(self.data[x\_col], self.data[y\_col])

            ax.set\_xlabel(x\_col)

            ax.set\_ylabel(y\_col)

        plt.tight\_layout()

        Exporter.export\_plot('output/MixedPlot.png')

        self.show()

**Папка src, файл exporter.py:**

import os

import matplotlib.pyplot as plt

class Exporter:

    @staticmethod

    def export\_plot(filename):

        base\_dir = os.path.join(os.path.dirname(\_\_file\_\_), "..")

        output\_dir = os.path.join(base\_dir, os.path.dirname(filename))

        if not os.path.exists(output\_dir):

            os.makedirs(output\_dir)

        plt.savefig(os.path.join(output\_dir, os.path.basename(filename)), format="png")

        print(f"Plot saved to {os.path.join(output\_dir, os.path.basename(filename))}")

        plt.show()

        plt.close()

**Файл main.py:**

from lab8.src.data\_loader import DataLoader

from lab8.src.analyzer import DataAnalyzer

from lab8.src.visualizer.line\_chart import LineChart

from lab8.src.visualizer.bar\_chart import BarChart

from lab8.src.visualizer.scatter\_plot import ScatterPlot

from lab8.src.visualizer.multi\_plot import MultiPlotVisualizer

from lab8.src.exporter import Exporter

def main():

    loader = DataLoader('lab8/data/dataset.csv')

    data = loader.load\_data()

    analyzer = DataAnalyzer(data)

    print("Extreme values:\n", analyzer.get\_extreme\_values())

    line\_chart = LineChart(data, x\_col='day', y\_col='temperature')

    line\_chart.plot()

    bar\_chart = BarChart(data, x\_col='day', y\_col='rain\_probability')

    bar\_chart.plot()

    scatter\_plot = ScatterPlot(data, x\_col='temperature', y\_col='rain\_probability')

    scatter\_plot.plot()

    multi\_plot = MultiPlotVisualizer(data)

    multi\_plot.plot\_subplots([

        ('line', 'day', 'temperature'),

        ('bar', 'day', 'rain\_probability'),

    ])

**Файл data.csv:**

day,temperature,rain\_probability

1,15,20

2,17,10

3,16,30

4,18,25

5,19,15

6,20,5

7,21,0

8,20,10

9,18,20

10,17,30

11,16,40

12,15,50

13,14,60

14,13,70

15,12,80

16,11,85

17,10,90

18,12,75

19,13,60

20,14,55

21,15,45

22,16,35

23,17,25

24,18,20

25,19,10

26,20,5

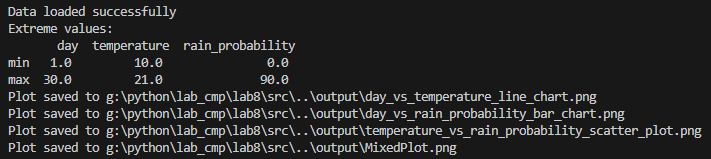
27,21,0

28,20,10

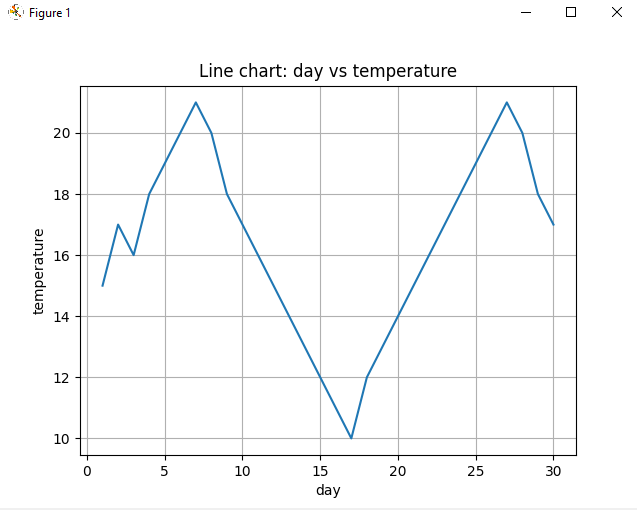
29,18,20

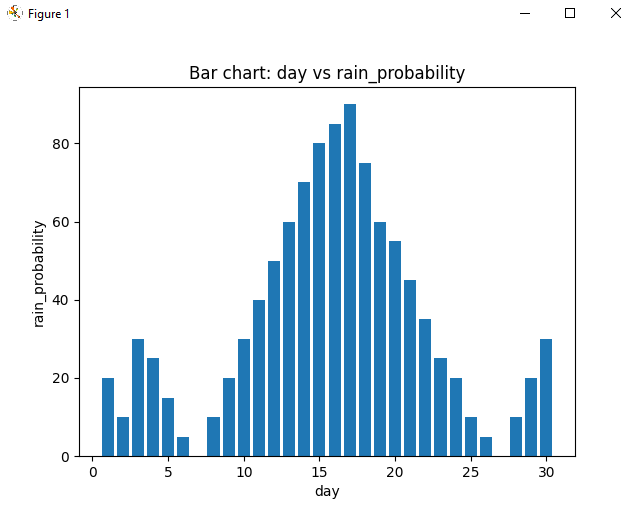
30,17,30

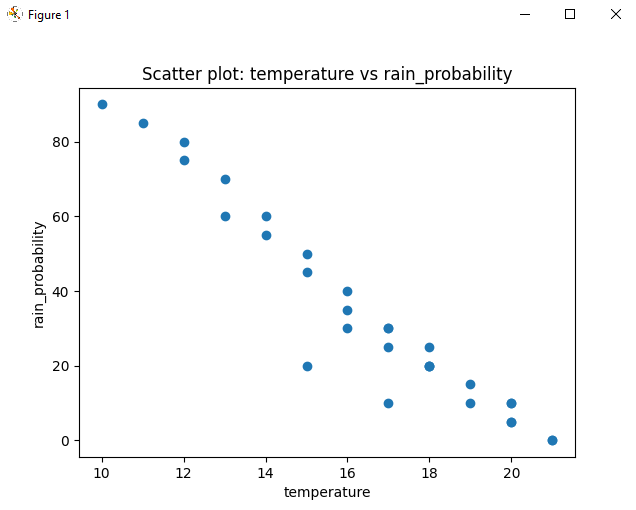
**Результат виконання:**

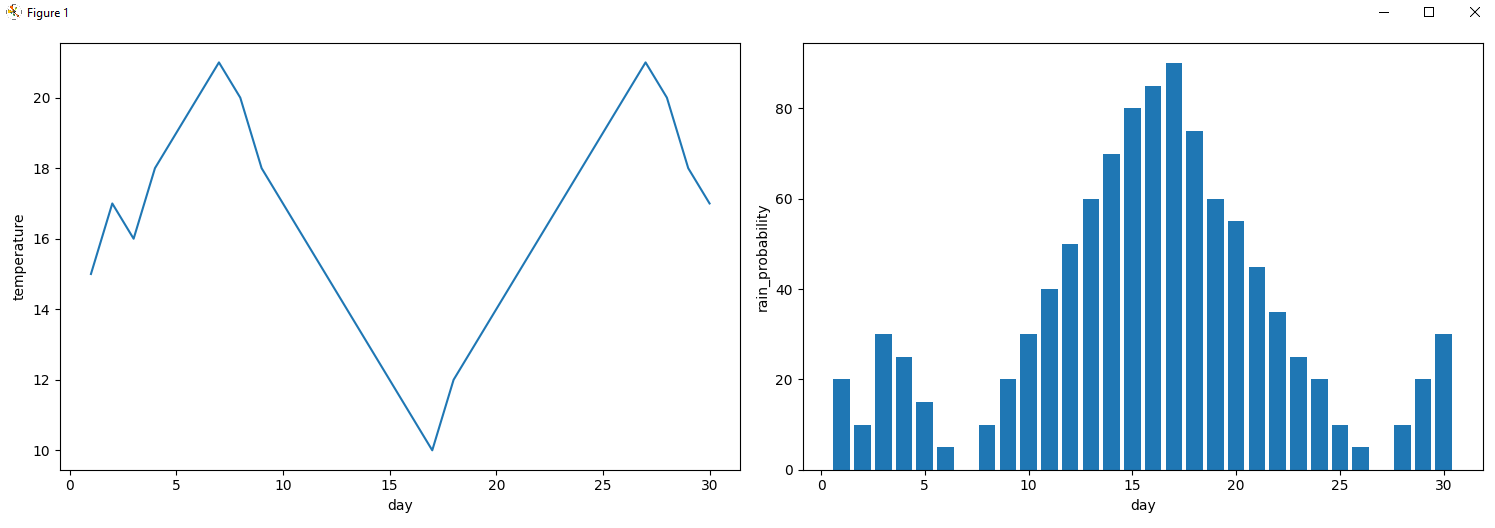
****

**Діаграми:**

****

****

****

****

**Висновок:** на цій лабораторній роботі я створив багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib