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:

Костюк В. О.

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 visualizers, файл bar\_chart.py:**

import matplotlib.pyplot as plt

from .base\_visualizer import BaseVisualizer

from lab8.export\_manager import ExportManager

class BarChartVisualizer(BaseVisualizer):

    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], label=f"{self.x\_col} vs {self.y\_col}")

        plt.xlabel(self.x\_col)

        plt.ylabel(self.y\_col)

        plt.title("Bar Chart")

        plt.legend()

        ExportManager.save\_plot\_as\_image("bar\_chart.png")

        plt.show()

**Папка visualizers, файл base\_visualizer.py:**

import matplotlib.pyplot as plt

class BaseVisualizer:

    def \_\_init\_\_(self, data):

        self.data = data

    def plot(self):

        raise NotImplementedError("Subclasses should implement this method")

**Папка visualizers, файл line\_chart.py:**import matplotlib.pyplot as plt

from .base\_visualizer import BaseVisualizer

from lab8.export\_manager import ExportManager

class LineChartVisualizer(BaseVisualizer):

    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.plot(self.data[self.x\_col], self.data[self.y\_col], label=f"{self.x\_col} vs {self.y\_col}")

        plt.xlabel(self.x\_col)

        plt.ylabel(self.y\_col)

        plt.title("Line Chart")

        plt.legend()

        ExportManager.save\_plot\_as\_image("line\_chart.png")

        plt.show()

**Папка visualizers, файл multi\_plot.py:**

import matplotlib.pyplot as plt

from .base\_visualizer import BaseVisualizer

from lab8.export\_manager import ExportManager

class MultiPlotVisualizer(BaseVisualizer):

    def \_\_init\_\_(self, data):

        super().\_\_init\_\_(data)

    def plot\_multiple(self, x\_col, y\_cols):

        fig, axs = plt.subplots(len(y\_cols), 1, figsize=(8, 6))

        for i, y\_col in enumerate(y\_cols):

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

            axs[i].set\_title(f"{x\_col} vs {y\_col}")

            axs[i].set\_xlabel(x\_col)

            axs[i].set\_ylabel(y\_col)

        fig.tight\_layout()

        ExportManager.save\_plot\_as\_image("multi\_plot.png")

        plt.show()

**Файл data\_loader.py:**

import pandas as pd

class DataLoader:

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

        self.\_file\_path = file\_path

        self.data = None

    def load\_data(self):

        self.data = pd.read\_csv(self.\_file\_path)

        print("Data loaded successfully.")

        return self.data

    def get\_extremes(self):

        # Визначення екстремальних значень для всіх числових стовпців

        return self.data.describe()

**Файл export\_manager.py:**

i import os

import matplotlib.pyplot as plt

class ExportManager:

    @staticmethod

    def save\_plot\_as\_image(filename):

        # Define the directory to save images

        output\_dir = "output\_images"

        os.makedirs(output\_dir, exist\_ok=True)  # Create folder if it doesn't exist

        # Full path for the image file

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

        # Save the plot to the specified directory

        plt.savefig(filepath)

        print(f"Saved plot as {filepath}")

**Файл main.py:**

from lab8.data\_loader import DataLoader

from lab8.visualizers.line\_chart import LineChartVisualizer

from lab8.visualizers.bar\_chart import BarChartVisualizer

from lab8.visualizers.multi\_plot import MultiPlotVisualizer

def main():

        # Load data

    data\_loader = DataLoader("lab8/data.csv")

    data = data\_loader.load\_data()

    # Display data extremes

    print("Data extremes:\n", data\_loader.get\_extremes())

    # Plot line chart

    line\_chart = LineChartVisualizer(data, x\_col="Date", y\_col="Value")

    line\_chart.plot()

    # Plot bar chart

    bar\_chart = BarChartVisualizer(data, x\_col="Category", y\_col="Date")

    bar\_chart.plot()

    # Plot multi-plot

    multi\_plot = MultiPlotVisualizer(data)

    multi\_plot.plot\_multiple(x\_col="Date", y\_cols=["Value", "Category"])

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

    main ()

**Файл data.csv:**

Date,Value,Category

2024-11-01,15,A

2024-11-02,16,A

2024-11-03,15,B

2024-11-04,14,B

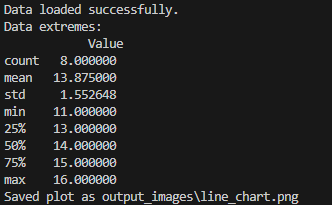
2024-11-05,13,C

2024-11-06,14,C

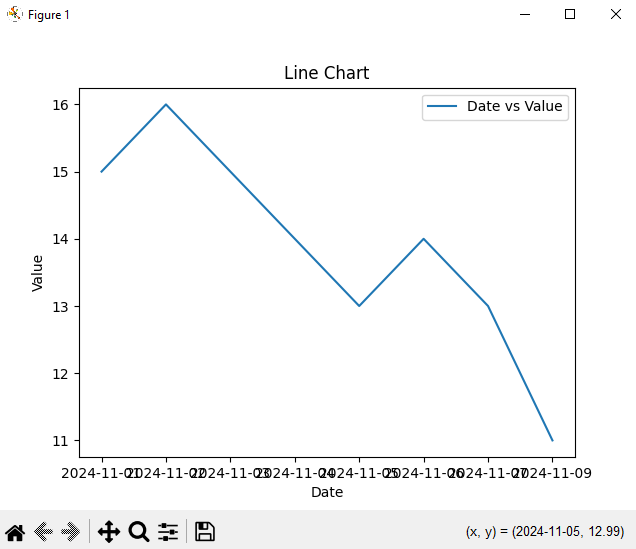
2024-11-07,13,A

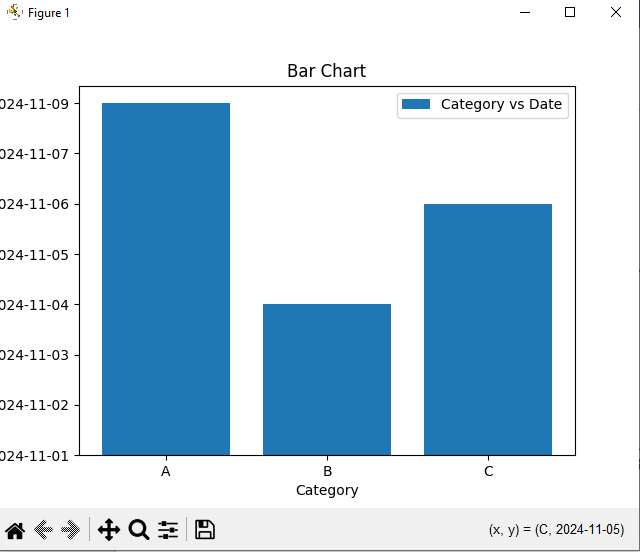
2024-11-09,11,A

**Результат виконання:**



**Діаграми:**







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