**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”**

**Виконав:**

**ст. гр. ІТ-32**

**Пазюк Сергій**

**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:**

import pandas as pd

import matplotlib.pyplot as plt

import json

import csv

import logging

from prettytable import PrettyTable

from consolemenu import ConsoleMenu

from consolemenu.items import FunctionItem

class Loader:

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

self.file\_path = file\_path

def load\_csv(self):

return pd.read\_csv(self.file\_path)

class Analyzer:

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

self.dataframe = dataframe

def find\_extremes(self, column):

return self.dataframe[column].max(), self.dataframe[column].min()

# Завдання 7: Розширені візуалізації

class Visualizer:

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

self.dataframe = dataframe

self.last\_figure = None

def plot\_basic(self, column, title, xlabel, ylabel):

if self.last\_figure is not None:

plt.close(self.last\_figure)

# Розділити малюнок на сітку 2x1 (2 рядки, 1 стовпець)

self.last\_figure, (ax1, ax2) = plt.subplots(nrows=2, ncols=1, figsize=(10, 8))

# Перший підграфік: лінійний графік

ax1.plot(self.dataframe.index, self.dataframe[column])

ax1.set\_title(title)

ax1.set\_xlabel(xlabel)

ax1.set\_ylabel(ylabel)

# Другий підграфік: стовпчикова діаграма

value\_counts = self.dataframe[column].value\_counts()

ax2.bar(value\_counts.index, value\_counts.values, color='skyblue', edgecolor='black')

ax2.set\_xlabel(xlabel)

ax2.set\_ylabel('Count')

# Налаштування розташування підграфіків

plt.tight\_layout()

plt.show()

def plot\_bar(self, categories, values, title='Bar Chart', xlabel='Category', ylabel='Value'):

self.last\_figure, ax = plt.subplots()

ax.bar(categories, values)

ax.set\_title(title)

ax.set\_xlabel(xlabel)

ax.set\_ylabel(ylabel)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

def plot\_histogram(self, column, bins=10):

self.last\_figure, ax = plt.subplots()

ax.hist(self.dataframe[column], bins=bins)

plt.show()

def plot\_scatter(self, column\_x, column\_y):

self.last\_figure, ax = plt.subplots()

ax.scatter(self.dataframe[column\_x], self.dataframe[column\_y])

plt.show()

def plot\_count\_by\_category(self, category):

counts = self.dataframe[category].value\_counts()

self.plot\_bar(counts.index, counts.values, title='Count of items by category', xlabel='Category', ylabel='Count')

def get\_figure(self):

return self.last\_figure

def save\_figure(self, filename):

if self.last\_figure:

self.last\_figure.savefig(filename)

class Exporter:

def export\_plot(self, figure, filename):

figure.savefig(filename)

class FileStorage:

@staticmethod

def store\_data(data, file\_format):

if file\_format == 'json':

with open('saved\_data.json', 'w') as file:

json.dump(data, file, indent=4)

elif file\_format == 'csv':

with open('saved\_data.csv', 'w', newline='') as file:

csv\_writer = csv.DictWriter(file, fieldnames=data[0].keys())

csv\_writer.writeheader()

csv\_writer.writerows(data)

elif file\_format == 'txt':

with open('saved\_data.txt', 'w') as file:

file.write(str(data))

# Завдання 1: Вибір CSV-набору даних

file\_path = "saved\_data.csv"

# Завдання 2: Завантаження даних з CSV

loader = Loader(file\_path)

dataframe = loader.load\_csv()

# Завдання 3: Дослідження даних

analyzer = Analyzer(dataframe)

max\_value, min\_value = analyzer.find\_extremes('completed')

print(f"Max Value: {max\_value}\nMin Value: {min\_value}")

# Завдання 4: Вибір типів візуалізацій

visualizer = Visualizer(dataframe)

# Завдання 6: Базова візуалізація

visualizer.plot\_basic('completed', 'Completion Status', 'Index', 'Completion Status')

# Завдання 9: Експорт і обмін

exporter = Exporter()

figure = visualizer.get\_figure()

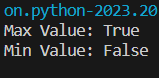
visualizer.save\_figure('plot.png')

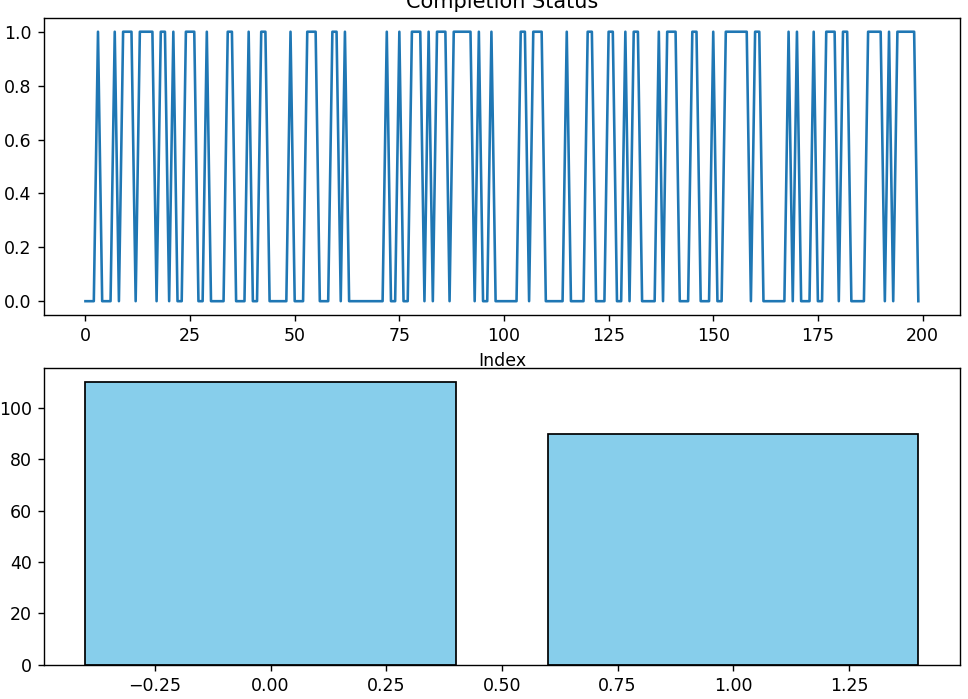
exporter.export\_plot(figure, 'plot.png')

file\_storage = FileStorage()

file\_storage.store\_data(dataframe, 'csv')

**Результат виконання программи**:





**Висновки:** Виконуючи ці завдання, я розробив додаток для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм)