Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Лабораторна робота №8

з навчальної дисципліни

«Спеціалізовані мови програмування»

на тему

«Візуалізація та обробка даних за допомогою

спеціалізованих бібліотек Python»

*Інститут комп’ютерних наук*

*та інформаційних технологій*

*Кафедри інформаційних*

*систем та мереж*

***Виконав:***

*Студент групи РІ-31*

*Олександр ЛУЧКЕВИЧ*

***Прийняла:***

*викладач*

*Cергій ЩЕРБАК*

*Львів – 2024*

**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)

**Код програми:**

Файл AdvancedVisualizer.py:

import matplotlib.pyplot as plt

from Lab\_8.BasicVisualizer import BasicVisualizer

class AdvancedVisualizer(BasicVisualizer):

def plot\_scatter(self, x\_column, y\_column):

plt.figure(figsize=(8, 6))

plt.scatter(self.data[x\_column], self.data[y\_column])

plt.xlabel(x\_column)

plt.ylabel(y\_column)

plt.title(f"{y\_column} vs {x\_column}")

plt.show()

def plot\_bar\_chart(self, category\_column, value\_column):

data\_grouped = self.data.groupby(category\_column)[value\_column].mean()

data\_grouped.plot(kind='bar', figsize=(10, 5))

plt.xlabel(category\_column)

plt.ylabel(f"Average {value\_column}")

plt.title(f"Average {value\_column} by {category\_column}")

plt.show()

Файл BasicVisualizer.py:

import matplotlib.pyplot as plt

import pandas as pd

class BasicVisualizer:

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

self.data = data

def plot\_line\_chart(self, x\_column, y\_column):

average\_price\_per\_year = self.data.groupby('Year\_of\_Manufacture\_\_c')['Sale\_Price\_\_c'].mean()

plt.figure(figsize=(10, 6))

plt.plot(average\_price\_per\_year.index, average\_price\_per\_year.values, marker='o', color='g', linestyle='-')

plt.xlabel(x\_column)

plt.ylabel(y\_column)

plt.grid(True)

plt.show()

Файл BasicVisualizer.py:

import re

class InputParser:

@staticmethod

def parse\_date(input\_str):

if re.match(r'^\d{4}-\d{2}-\d{2}$', input\_str):

return "Дата розпізнана!"

return "Невірний формат дати"

@staticmethod

def parse\_phone(input\_str):

if re.match(r'^\+?\d[\d -]{7,12}\d$', input\_str):

return "Телефонний номер розпізнано!"

return "Невірний формат телефону"

Файл InputParser.py:

import matplotlib.pyplot as plt

import pandas as pd

class BasicVisualizer:

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

self.data = data

def plot\_line\_chart(self, x\_column, y\_column):

average\_price\_per\_year = self.data.groupby('Year\_of\_Manufacture\_\_c')['Sale\_Price\_\_c'].mean()

plt.figure(figsize=(10, 6))

plt.plot(average\_price\_per\_year.index, average\_price\_per\_year.values, marker='o', color='g', linestyle='-')

plt.xlabel(x\_column)

plt.ylabel(y\_column)

plt.grid(True)

plt.show()

Файл MultiPlotVisualizer.py:

import matplotlib.pyplot as plt

from Lab\_8.BasicVisualizer import BasicVisualizer

class MultiPlotVisualizer(BasicVisualizer):

def plot\_multiple(self):

fig, axs = plt.subplots(2, 2, figsize=(12, 10))

average\_price\_per\_year = self.data.groupby('Year\_of\_Manufacture\_\_c')['Sale\_Price\_\_c'].mean()

axs[0, 0].plot(average\_price\_per\_year.index, average\_price\_per\_year.values, marker='o', linestyle='-')

axs[0, 0].set\_title('Sale Price Over Year')

axs[0, 1].scatter(self.data['Mileage\_\_c'], self.data['Sale\_Price\_\_c'])

axs[0, 1].set\_title('Mileage vs Sale Price')

self.data['Condition\_\_c'].value\_counts().plot(kind='bar', ax=axs[1, 0], color=['blue', 'orange'])

axs[1, 0].set\_title('Condition Distribution')

self.data['Fuel\_Type\_\_c'].value\_counts().plot(kind='pie', ax=axs[1, 1], autopct='%1.1f%%')

axs[1, 1].set\_title('Fuel Type Distribution')

plt.tight\_layout()

plt.show()

Файл main.py:

from DAL.DataAnalyzer import DataAnalyzer

from DAL.DataLoader import DataLoader

from DAL.DataPreparer import DataPreparer

from Lab\_8.AdvancedVisualizer import AdvancedVisualizer

from Lab\_8.BasicVisualizer import BasicVisualizer

from Lab\_8.MultiPlotVisualizer import MultiPlotVisualizer

from UI.MenuItem import MenuItem

from UI.MenuBuilder import MenuBuilder

class VisualizationApp:

def \_\_init\_\_(self, filepath: str):

self.filepath = filepath

self.prepared\_data = self.\_prepare\_data()

def \_prepare\_data(self):

print("Завантаження даних...")

loader = DataLoader(self.filepath)

data = loader.get\_data()

print("Аналіз даних...")

analyzer = DataAnalyzer(data)

print("Екстремальні значення:")

print(analyzer.get\_extreme\_values())

print("Підготовка даних для візуалізації...")

preparer = DataPreparer(data)

return preparer.prepare\_data\_for\_visualization()

def plot\_line\_chart(self):

visualizer = BasicVisualizer(self.prepared\_data)

visualizer.plot\_line\_chart('Mileage\_\_c', 'Sale\_Price\_\_c')

def plot\_scatter\_and\_bar(self):

visualizer = AdvancedVisualizer(self.prepared\_data)

visualizer.plot\_scatter('Mileage\_\_c', 'Sale\_Price\_\_c')

visualizer.plot\_bar\_chart('Engine\_Power\_\_c', 'Sale\_Price\_\_c')

def plot\_multiple(self):

visualizer = MultiPlotVisualizer(self.prepared\_data)

visualizer.plot\_multiple()

def run():

filepath = "C:/Users/Користувач/Desktop/5 семестр/1/SMP/Source/CarsCSV.csv"

app = VisualizationApp(filepath)

menu\_items = [

MenuItem("1", "Побудувати лінійний графік", app.plot\_line\_chart),

MenuItem("2", "Побудувати точковий графік і гістограму", app.plot\_scatter\_and\_bar),

MenuItem("3", "Побудувати кілька графіків", app.plot\_multiple),

MenuItem("0", "Вийти", exit),

]

menu = MenuBuilder(menu\_items)

while True:

menu.initialize()

**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