МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №8

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав студент

групи ІТ-21сп

**Гузар А.І.**

Прийняв

**Щербак С.С.**

Львів - 2023

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

**Завдання 1:** ВибірCSV**-**наборуданих

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)

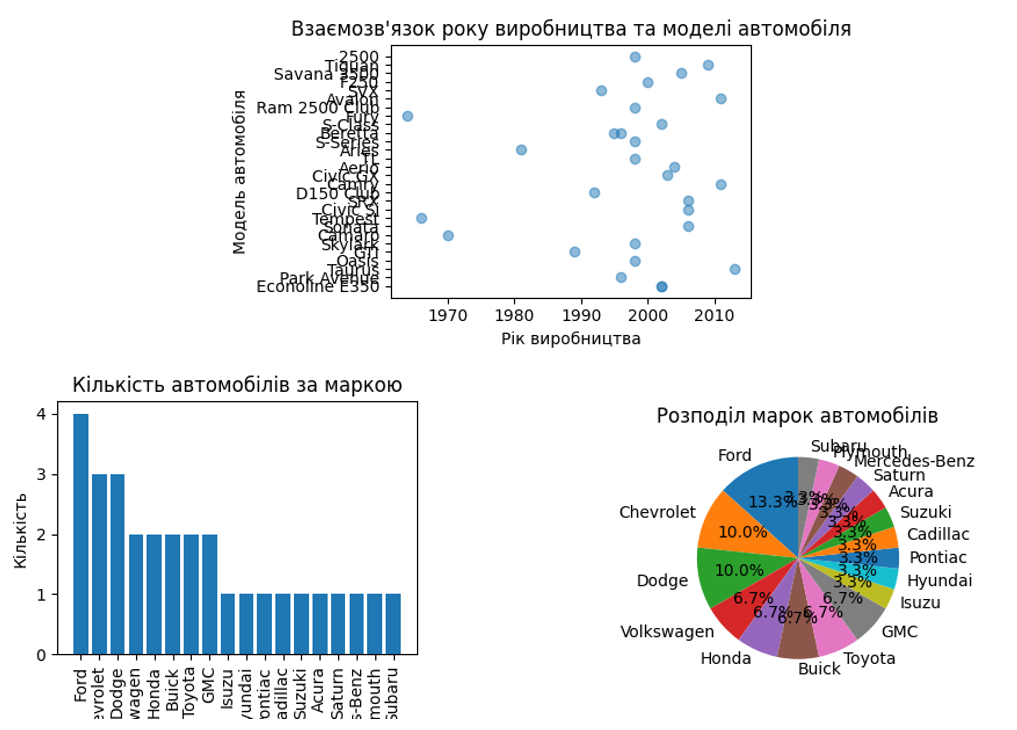
Клас main.py:

from Lab8.csv\_visual import DataVisualizer  
  
def main():  
 #Шлях до CSV-файлу  
 csv\_file\_path = E:\labo4ki\Labs\Lab8\data\mock\_data.csv'  
  
 while True:  
 #Створення об'єкту DataVisualizer  
 data\_visualizer = DataVisualizer(csv\_file\_path)  
  
 #Завантаження та дослідження даних  
 data\_visualizer.explore\_data()  
  
 #Запит користувача щодо вибору діаграми  
 user\_choice = input("1. Кругова діаграма \n2. Діаграма розсіювання \n3. Стовпцева діаграма \n4. Вивести всі діаграми \nВведіть номер діаграми (1-4) або 'x' для виходу: ")  
  
 if user\_choice.lower() == 'x':  
 break  
  
 #Візуалізація даних згідно з вибором користувача  
 data\_visualizer.visualize\_data(user\_choice)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Клас csv\_visual.py:

import pandas as pd  
import matplotlib.pyplot as plt  
import plotly.express as px  
  
class DataVisualizer:  
 def \_\_init\_\_(self, csv\_file\_path):  
 self.df = pd.read\_csv(csv\_file\_path)  
  
 def explore\_data(self):  
 #Екстремальні значення по стовпцях  
 min\_values = self.df.min()  
 max\_values = self.df.max()  
  
 print("Мінімальні значення:")  
 print(min\_values)  
  
 print("\nМаксимальні значення:")  
 print(max\_values)  
  
 def visualize\_data(self, user\_choice):  
 if user\_choice == "1":  
 self.pie\_chart()  
 elif user\_choice == "2":  
 self.scatter\_plot()  
 elif user\_choice == "3":  
 self.bar\_chart()  
 elif user\_choice == "4":  
 self.show\_all\_plots()  
 else:  
 print("Невірний вибір. Будь ласка, введіть номер від 1 до 4.")  
  
 def pie\_chart(self):  
 #Візуалізація частки кожної марки автомобіля  
 car\_make\_distribution = self.df['car\_make'].value\_counts()  
 car\_make\_distribution.plot(kind='pie', autopct='%1.1f%%', title='Розподіл марок автомобілів')  
 plt.show()  
  
 def scatter\_plot(self):  
 #Візуалізація взаємозв'язку року виробництва та моделі автомобіля  
 plt.scatter(self.df['car\_model\_year'], self.df['car\_model'], alpha=0.5)  
 plt.xlabel('Рік виробництва')  
 plt.ylabel('Модель автомобіля')  
 plt.title('Взаємозв\'язок року виробництва та моделі автомобіля')  
 plt.show()  
  
 def bar\_chart(self):  
 #Візуалізація кількості автомобілів за маркою  
 car\_make\_counts = self.df['car\_make'].value\_counts()  
 car\_make\_counts.plot(kind='bar', xlabel='Марка автомобіля', ylabel='Кількість',  
 title='Кількість автомобілів за маркою', rot=45)  
 plt.show()  
  
 def show\_all\_plots(self):  
 # Вивести всі три графіки  
 fig, axes = plt.subplots(2, 2, figsize=(10, 8))  
  
 # Кругова діаграма  
 car\_make\_distribution = self.df['car\_make'].value\_counts()  
 axes[0, 0].pie(car\_make\_distribution, autopct='%1.1f%%', labels=car\_make\_distribution.index, startangle=90)  
 axes[0, 0].set\_title('Розподіл марок автомобілів')  
 #Діаграма розсіювання  
 axes[0, 1].scatter(self.df['car\_model\_year'], self.df['car\_model'], alpha=0.5)  
 axes[0, 1].set\_xlabel('Рік виробництва')  
 axes[0, 1].set\_ylabel('Модель автомобіля')  
 axes[0, 1].set\_title('Взаємозв\'язок року виробництва та моделі автомобіля')  
 #Стовпцева діаграма  
 car\_make\_counts = self.df['car\_make'].value\_counts()  
 axes[1, 0].bar(car\_make\_counts.index, car\_make\_counts)  
 axes[1, 0].set\_xlabel('Марка автомобіля')  
 axes[1, 0].set\_ylabel('Кількість')  
 axes[1, 0].set\_title('Кількість автомобілів за маркою')  
 axes[1, 0].tick\_params(axis='x', labelrotation=90) # Обертання тексту на горизонтальних стовпчиках  
 plt.delaxes(axes[1, 1])  
 #Встановлення відстані між графіками  
 plt.subplots\_adjust(wspace=0.5, hspace=0.5)  
 #Збереження графіків у форматах PNG та HTML  
 plt.savefig(E:\labo4ki\Labs\Lab8\data\output\_plot.png')  
 #Використання Plotly для збереження графіка у форматі HTML  
 fig = px.scatter(self.df, x='car\_model\_year', y='car\_model',  
 title='Взаємозв\'язок року виробництва та моделі автомобіля')  
 fig.write\_html(E:\labo4ki\Labs\Lab8\data\output\_plot.html')  
 plt.show()

На рисунку 1 зображено результат виконання програми.



*Рис1. Результат запиту*

**Висновок:** під час виконання лабораторної роботи я навчився розробляти додатки для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП.