**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №4**

**з навчальної дисципліни «Технології Data Science»**

**Тема:**

**РЕАЛІЗАЦІЯ ПРОЦЕСІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ: МІНІ**

**ПРОЕКТИ В ГАЛУЗЯХ OLAP, Data Mining, Text Mining, Voice Recognition**

**Виконав:**

Студент 4 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Трикош І. В.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

## **І. Мета:**

Виявити дослідити та узагальнити особливості інтелектуального аналізу даних та технологій OLAP, Data Mining, Text Mining, Voice Recognition.

## **ІІ. Завдання:**

1. Лабораторія провідної ІТ-компанії реалізує проект із розробки ERP системи для автоматизації процесів управління у сфері завдань електронної комерції. Вам, як Data Science Engineer поставлене наступне завдання. Розробити програмний скрипт мовою Python що реалізує обчислювальний алгоритм ERP-системи із впровадженням технології інтелектуального аналізу даних відповідно до технічних умов:

**ІІ рівень складності 9 балів, викладених у табл.2 – R&D міні проект.**

Варіант 9:

Розробити програмний скрипт, що реалізує збір і узагальнення інформації про меню найбільш популярних кав’ярень м. Львів.

Обрані кав’ярні:

Львівська копальня кави: <https://kopalnia.choiceqr.com/section:menyu-v-kav-yarnya>

Вірменка: <https://virmenka.com.ua/uk/menu/>

Світ кави: <https://svitkavy.com/uk/cafes/menu/katedralna/>

## **ІІІ. Результати виконання лабораторної роботи.**

* 1. **Синтезована математична модель**

Відповідно до умов задачі синтезовано математичну модель, що реалізує збір та узагальнення інформації про меню найбільш популярних кав’ярень м. Львів.

Збір інформації відбувається за допомогою парсингу сторінки меню відповідної кав’ярні. Отримання сторінки відбувається з допомогою ChromeDriver, а сам парсинг – з допомогою BeautifulSoup.

Узагальнення інформації:

* знаходиться кількість напоїв та страв у кожному меню та у всіх меню разом.
* знаходиться середня ціна меню по кожній кав’ярні та по усіх кав’ярнях разом за формулою:

де n – кількість напоїв та страв у меню, price – ціна за i-тий товар або напій.

* ціни розподіляються за категоріями Низька (до 100 грн), Середня (від 100 до 200 грн), Висока (від 200 грн).
* порівняння цін на напої та страви, що є у всіх меню.
  1. **Результати архітектурного проектування та їх опис**

На рис. 1 зображена блок-схема до програмного скрипту:



Рисунок 1 – Блок схема алгоритму

Робота алгоритму розпочинається з ініціалізації початкових параметрів (url та назви кав’ярень). Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.1.

У блоці 2 виконується парсинг даних із сайтів та збереження імен та цін у словник.

У блоці 3 дані очищаються та зберігаються у відповідні датафрейми.

У блоці 4 виконується збереження датафреймів у csv файли.

У блоці 5 відбувається обчислення середньої ціни у меню у кожній кав’ярні.

У блоці 6 знаходиться середня ціна у меню у всіх кав’ярнях.

У блоці 7 відбувається розподіл цін по категоріях Низька, Середня, Висока.

У блоці 8 знаходиться розмір кожного меню окремо та їх разом узятих.

У блоці 9 відбувається порівняння цін на напої та страви, які є у всіх меню відразу. На цьому виконання алгоритму завершене.

* 1. **Опис структури проекту програми**

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

Проєкт базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

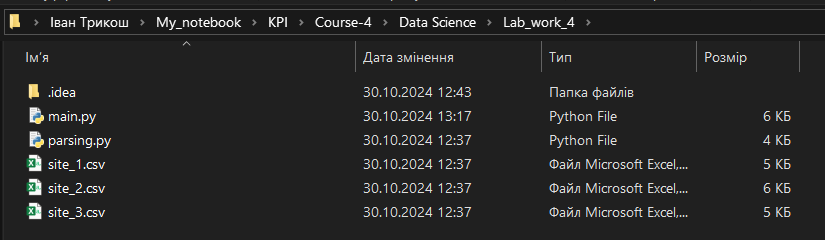


Рисунок 2 – Структура проєкту

Lab\_work\_4 – головний каталог проєкту

main.py – файл головного програмного коду

parsing.py – файл програмного коду для парсингу сайтів кав’ярень

site\_1.csv, site\_2.csv, site\_3.csv – результати парсингу для сайтів кав’ярень

* 1. **Результати роботи програми відповідно до завдання (допускається у формі скриншотів)**

Результатом роботи програми є сукупність послідовності графічних вікон, що реалізують умови завдання лабораторної роботи.

1. На рисунку 3 зображено результати парсингу меню сайтів кав’ярень:

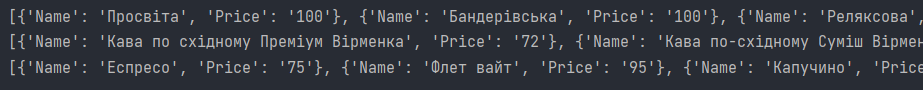


Рисунок 3 – Отримані дані про меню кав’ярень

1. На рисунках 4-8 зображено узагальнення даних про меню кав’ярень:

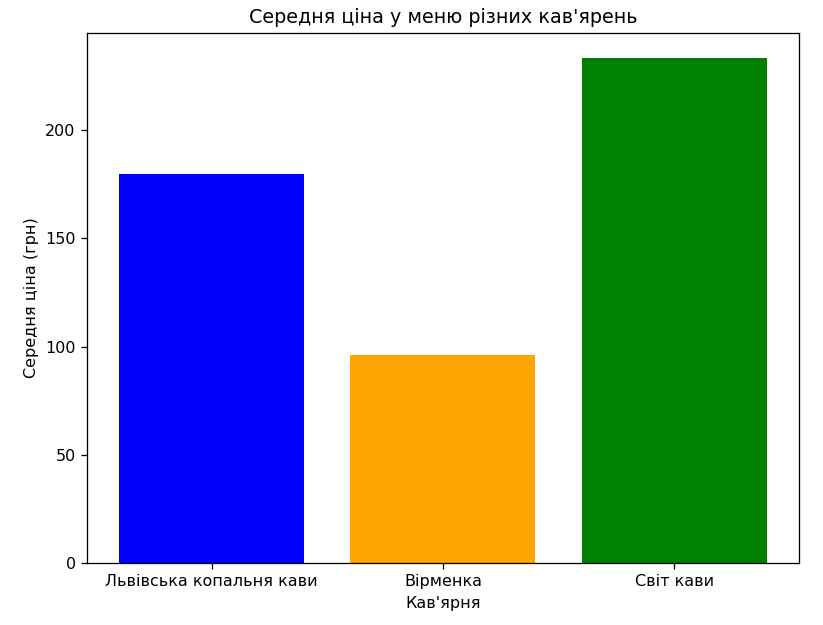


Рисунок 4 – Середня ціна у меню різних кав’ярень



Рисунок 5 – Середня ціна по усіх трьох кав’ярнях

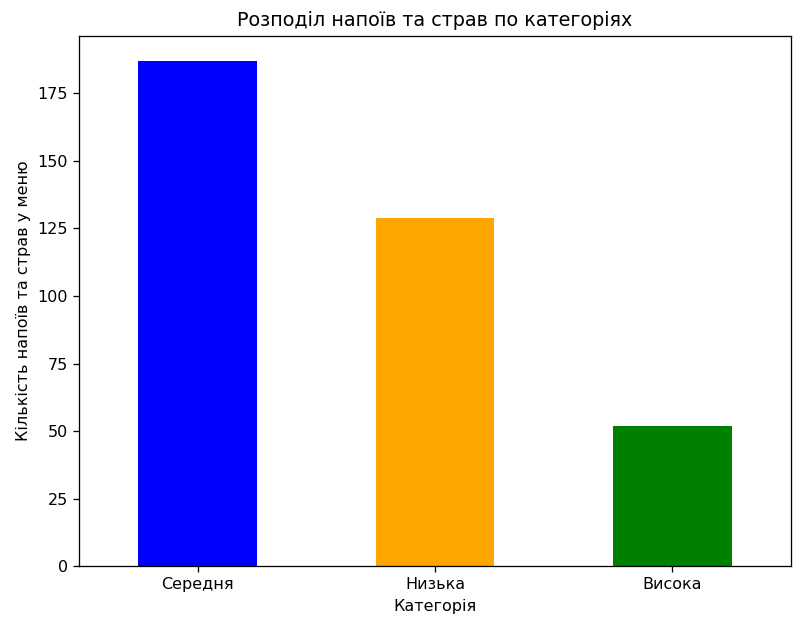


Рисунок 6 – Розподіл напоїв та страв по цінових категоріях

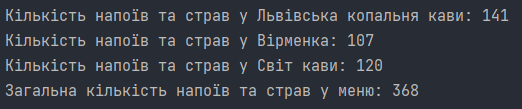


Рисунок 7 – Кількість напоїв та страв у кожній кав’ярні та у всіх кав’ярнях разом

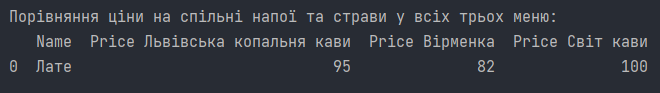


Рисунок 8 – Порівняння ціни на напої та страви, які є у кожному меню

Бачимо, що кав’ярня з найдешевшими цінами (і за середньої ціною, і за ціною лате) – Вірменка, проте також вона має найменший асортимент. Львівська копальня кави є найбільшою за асортиментом та має середні ціни. Світ кави є найдорожчою за цінами кав’ярнею.

* 1. **Програмний код, що забезпечує отримання результату (допускається у формі скриншотів)**

Програмний код послідовно реалізує алгоритм на рис.1 та спрямований на отримання результатів, поданих на рис.3-8.

Для спрощення програмного коду і раціоналізації обчислень застосовано функціональні механізми створення підпрограм.

При цьому використано можливості Python бібліотек: pandas, matplotlib, beautifulsoup, selenium.

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

parsing.py

*"""  
Парсинг меню із сайтів кав'ярень Львова  
URL сайтів:  
 - https://kopalnia.choiceqr.com/section:menyu-v-kav-yarnya  
 - https://virmenka.com.ua/uk/menu/  
 - https://svitkavy.com/uk/cafes/menu/katedralna/  
"""  
  
from* bs4 *import* BeautifulSoup  
*from* selenium *import* webdriver  
*import* pandas *as* pd  
  
driver = webdriver.Chrome()  
  
  
*def* fetch\_html(url):  
 *""" Отримання HTML коду сторінки за URL адресою """* driver.get(url)  
 driver.implicitly\_wait(2)  
 response = driver.page\_source  
 *return* response  
  
  
*def* parse\_site\_1():  
 *""" Парсинг першого сайту """* url = "https://kopalnia.choiceqr.com/section:menyu-v-kav-yarnya"  
 html = fetch\_html(url)  
 soup = BeautifulSoup(html, "html.parser")  
 items = soup.select(".styles\_menu-item-left\_\_HxAVf")  
 data = []  
 *for* item *in* items:  
 name = item.select\_one(".styles\_menu-item-title\_\_Mnuv\_").get\_text(strip=*True*)  
 price = (item.select\_one(".styles\_menu-item-price\_\_G8nZ\_ .styles\_discount\_\_EE8JM").get\_text(strip=*True*).split(' '))[0]  
 data.append({"Name": name, "Price": price})  
 *return* data  
  
  
*def* parse\_site\_2():  
 *""" Парсинг другого сайту """* url = "https://virmenka.com.ua/uk/menu/"  
 html = fetch\_html(url)  
 soup = BeautifulSoup(html, "html.parser")  
 items = soup.select(".menu-section\_\_item")  
 data = []  
 *for* item *in* items:  
 name = item.select\_one(".menu-item\_\_title").get\_text(strip=*True*)  
 price = item.select\_one(".menu-item\_\_price").get\_text(strip=*True*).split(' ')[0]  
 data.append({"Name": name, "Price": price})  
 *return* data  
  
  
*def* parse\_site\_3():  
 *""" Парсинг третього сайту """* url = "https://svitkavy.com/uk/cafes/menu/katedralna/"  
 html = fetch\_html(url)  
 soup = BeautifulSoup(html, "html.parser")  
 items = soup.select("div.cafe-menu ul")  
 names = soup.select("div.cafe-menu h2")[1:]  
 data = []  
 *for* item\_index *in* range(len(items)):  
 lis = items[item\_index].select("li")  
 *for* li *in* lis:  
 name = li.select\_one(".item").contents[0].get\_text(strip=*True*)  
 *if* name[0].islower():  
 name = names[item\_index].get\_text(strip=*True*) + " " + name  
 price = li.select\_one(".price").get\_text(strip=*True*).split('/')[0]  
 data.append({"Name": name, "Price": price})  
 *return* data  
  
  
*# Парсинг першого меню*menu = parse\_site\_1()  
print(menu)  
*# Збереження першого меню*df = pd.DataFrame(menu)  
df.to\_csv('site\_1.csv', index=*False*, encoding='utf-8-sig')  
  
*# Парсинг другого меню*menu = parse\_site\_2()  
print(menu)  
*# Збереження другого меню*df = pd.DataFrame(menu)  
df.to\_csv('site\_2.csv', index=*False*, encoding='utf-8-sig')  
  
*# Парсинг третього меню*menu = parse\_site\_3()  
print(menu)  
*# Збереження третього меню*df = pd.DataFrame(menu)  
df.to\_csv('site\_3.csv', index=*False*, encoding='utf-8-sig')  
  
driver.quit()

main.py

*"""  
Узагальнення інформації про меню кав'ярень Львова:  
 - знаходження середньої ціни напоїв або страв у меню кожної кав'ярні  
 - знаходження середньої ціни напоїв або страв у меню усіх кав'ярень  
 - розподіл цін напоїв та страв по категоріях Низька, Середня, Висока по всіх кав'ярнях  
 - кількість напоїв та страв у меню кожної кав'ярні  
 - порівняння ціни напоїв та страв, які є у кожній із кав'ярень  
"""  
  
import* pandas *as* pd  
*from* matplotlib *import* pyplot *as* plt  
  
*# Зчитування даних про меню кав'ярень*df\_site1 = pd.read\_csv("site\_1.csv")  
df\_site2 = pd.read\_csv("site\_2.csv")  
df\_site3 = pd.read\_csv("site\_3.csv")  
  
*# Назви кав'ярень*site\_1\_name = "Львівська копальня кави"  
site\_2\_name = "Вірменка"  
site\_3\_name = "Світ кави"  
  
  
*# Обчислення середньої ціни у меню кожної кав'ярні*site\_means = {  
 site\_1\_name: df\_site1["Price"].mean(),  
 site\_2\_name: df\_site2["Price"].mean(),  
 site\_3\_name: df\_site3["Price"].mean()  
}  
print("Середня ціна у кожній кав'ярні (грн): ", site\_means)  
  
*# Побудова стовпчастої діаграми*plt.figure(figsize=(8, 6))  
plt.bar(list(site\_means.keys()), list(site\_means.values()), color=["blue", "orange", "green"])  
plt.xlabel("Кав'ярня")  
plt.ylabel("Середня ціна (грн)")  
plt.title("Середня ціна у меню різних кав'ярень")  
plt.show()  
  
  
*# Обчислення середньої ціни у меню у всіх кав'ярнях  
# Дані про ціни з трьох кав'ярень*prices = pd.concat([df\_site1["Price"], df\_site2["Price"], df\_site3["Price"]])  
print("\nСередня ціна по трьох кав'ярнях (грн): ", prices.mean())  
  
  
*# Визначення розподілу цін напоїв та страв по категоріях Низька, Середня, Висока по всіх кав'ярнях  
def* price\_category(price):  
 *""" Визначення категорії за значенням ціни """  
 if* price < 100:  
 *return* "Низька"  
 *elif* 100 <= price <= 200:  
 *return* "Середня"  
 *else*:  
 *return* "Висока"  
  
  
*# Визначаємо категорії*prices\_category = prices.apply(price\_category)  
print("\nРозподіл по категоріях:\n", prices\_category.value\_counts())  
*# Побудова гістограми*plt.figure(figsize=(8, 6))  
prices\_category.value\_counts()\  
 .plot(kind="bar", color=["blue", "orange", "green"])  
plt.xlabel("Категорія")  
plt.ylabel("Кількість напоїв та страв у меню")  
plt.title("Розподіл напоїв та страв по категоріях")  
plt.xticks(rotation=0)  
plt.show()  
  
  
*# Знаходження кількості напоїв та страв у кожному меню та у всіх разом  
# Підрахунок кількості напоїв та страв у кожному меню*count\_site1 = df\_site1.shape[0]  
count\_site2 = df\_site2.shape[0]  
count\_site3 = df\_site3.shape[0]  
  
*# Загальна кількість напоїв та страв у всіх меню*total\_count = count\_site1 + count\_site2 + count\_site3  
  
print()  
print(f"Кількість напоїв та страв у {site\_1\_name}: {count\_site1}")  
print(f"Кількість напоїв та страв у {site\_2\_name}: {count\_site2}")  
print(f"Кількість напоїв та страв у {site\_3\_name}: {count\_site3}")  
print(f"Загальна кількість напоїв та страв у меню: {total\_count}")  
  
  
*# Порівняння цін на напої та страви, які є у кожній з кав'ярень  
# Множини назв напоїв та страв у кожному меню*site1\_items = set(df\_site1["Name"])  
site2\_items = set(df\_site2["Name"])  
site3\_items = set(df\_site3["Name"])  
  
*# Знаходимо спільні товари у всіх трьох меню*common\_items = site1\_items & site2\_items & site3\_items  
  
*# Фільтруємо кожне меню для отримання лише спільних напоїв та страв*df\_site1\_common = df\_site1[df\_site1["Name"].isin(common\_items)]  
df\_site2\_common = df\_site2[df\_site2["Name"].isin(common\_items)]  
df\_site3\_common = df\_site3[df\_site3["Name"].isin(common\_items)]  
  
*# Об'єднуємо для порівняння*comparison\_df = pd.merge(  
 df\_site1\_common[["Name", "Price"]], df\_site2\_common[["Name", "Price"]],  
 on="Name", suffixes=(f" {site\_1\_name}", f" {site\_2\_name}")  
)  
comparison\_df = pd.merge(comparison\_df, df\_site3\_common[["Name", "Price"]], on="Name")  
comparison\_df.rename(columns={"Price": f"Price {site\_3\_name}"}, inplace=*True*)  
  
print("\nПорівняння ціни на спільні напої та страви у всіх трьох меню:")  
print(comparison\_df)

## **IV. Висновки.**

У ході виконання лабораторної роботи було розроблено програмний скрипт, що реалізує збір і узагальнення інформації про меню найбільш популярних кав’ярень м. Львів.

Виконав: студент Трикош І. В.