**Міністерство освіти і науки України**

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

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №9**

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

**Тема:**

**МАКЕТ GIS СИСТЕМИ**

**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:**

Дослідити, виявити та узагальнити особливості впровадження геоінформаційних технологій в процесах Data Science.

## **ІІ. Завдання:**

В інтересах потенційних замовників – державних та комерційних установ R&D лабораторія провідної ІТ-компанії розробляє макет GIS системи. Необхідно розробити програмний скрипт Backend – компоненту GIS системи з функціоналом, за рівнями складності.

**ІІ рівень складності 8 балів.**

Відповідно до технічних умов, табл.1 додатку. Регіон – місто Київ.



\*\*\* додаткова пропозиція + 2 бали – реалізувати GIS візуалізацію аналітики даних, що використовувались Вами в межах цього курсу, якщо вони мають геопросторові атрибути / ознаки. Для виконання додаткового завдання візуалізуємо кав’ярні Львова (ЛР4).

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

* 1. **Синтезована математична модель**

Геоінформаційна система (GIS) – це система для збору, зберігання, аналізу, управління і візуалізації географічно пов’язаної інформації. Може містити як геопросторові (координати), так і атрибутивні (додаткова інформація про об’єкт) дані. Вона широко застосовується в урбаністиці, екології, логістиці, сільському господарстві тощо.

Відповідно до умов задачі синтезовано математичну модель, що визначає, аналізує та геопросторово візуалізує інформацію про медичні заклади Києва, і візуалізує кав’ярні Львова на карті.

Отримання інформації про медичні заклади і кав’ярні Києва та Львова відповідно отримаємо з допомогою сервісу Open Street Maps, а саме його бібліотеки osmnx для Python. Для отримання району Києва, у якому розміщений відповідний медзаклад, скористаємось бібліотекою geopy, а саме її геокодером Nominatim. Щоб відобразити заклади на карті, скористаємось бібліотекою folium.

* 1. **Результати архітектурного проектування та їх опис**

На рис. 1 зображена блок-схема до програмного скрипту №1:



Рисунок 1 – Блок схема алгоритму №1

Робота алгоритму №1 розпочинається із визначення початкових параметрів, таких як місто (Київ) та тип закладів (госпіталь та клініка) і завантаження інформації з допомогою osmnx. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.1.

У блоці 2 відбувається очищення даних. Сюди входить вибір основних стовпців, додавання стовпців з широтою та довготою і видалення рядків, що містять пропущені дані.

У блоці 3 виконується підрахунок кількості закладів за типом (госпіталь або клініка).

У блоці 4 підраховується кількість закладів по районах Києва. Для цього спочатку визначається район Києва за координатами медзакладу з допомогою геокодера Nominatim.

У блоці 5 виконується відображення медзакладів на карті з допомогою folium. На цьому виконання алгоритму завершене.

На рис. 2 зображена блок-схема до програмного скрипту №2:



Рисунок 2 – Блок схема алгоритму №2

Робота алгоритму №2 розпочинається із визначення початкових параметрів, таких як місто (Львів) та тип закладів (кав’ярня) і завантаження інформації з допомогою osmnx. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.2.

У блоці 2 відбувається очищення даних. Сюди входить вибір основних стовпців і видалення рядків, що містять пропущені дані.

У блоці 3 виконується відображення кав’ярень на карті з допомогою folium. На цьому виконання алгоритму завершене.

* 1. **Опис структури проекту програми**

Для реалізації розроблених алгоритмів мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

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

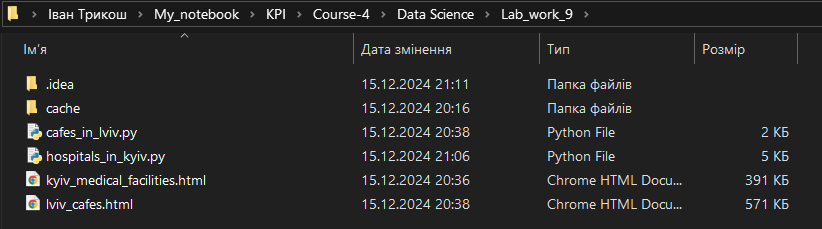


Рисунок 3 – Структура проєкту

Lab\_work\_9 – головний каталог проєкту

cache – містить кешовані дані про заклади (створені автоматично)

cafes\_in\_lviv.xlsx – файл програмного коду №2

hospitals\_in\_kyiv.py – файл програмного коду №1

kyiv\_medical\_facillities.html – медзаклади Києва на карті (результат скрипту №1)

lviv\_cafes.html – кав’ярні Львова на карті (результат скрипту №2)

* 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 рисунках 4-10 зображено результати роботи алгоритму №1:

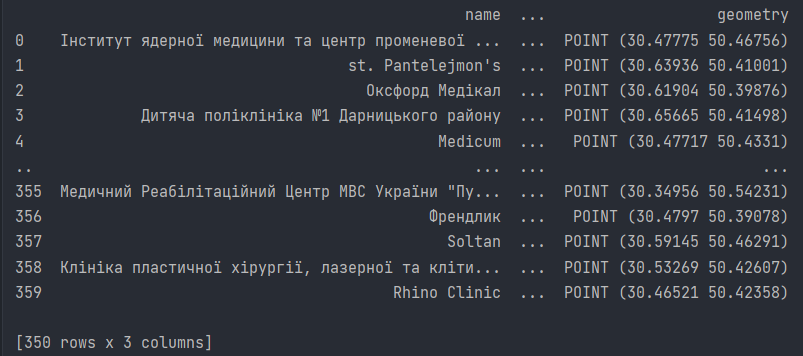


Рисунок 4 – Дані про медзаклади Києва

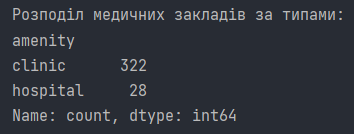


Рисунок 5 – Розподіл медичних закладів за типами (клініка та госпіталь)

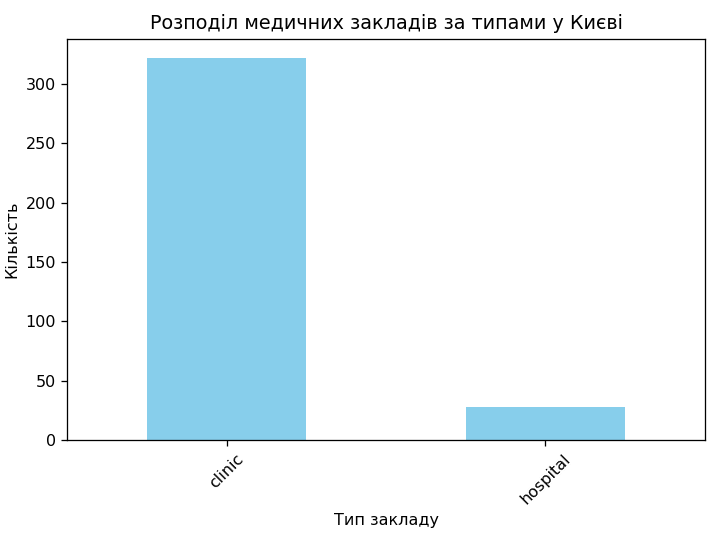


Рисунок 6 – Графік розподілу медзакладів за типами

Бачимо, що кількість клінік значно переважає кількість госпіталів.

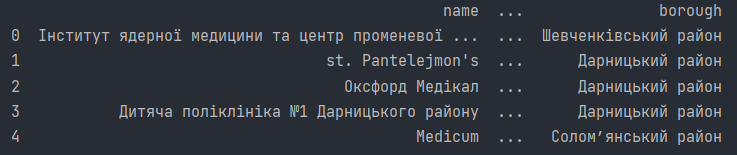


Рисунок 7 – Медзаклади і райони їх розміщення

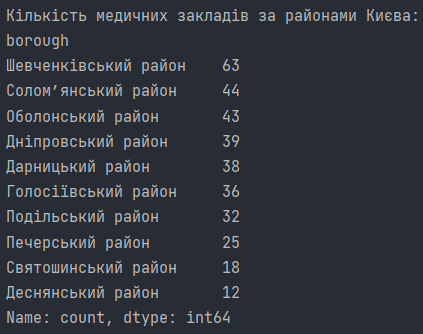


Рисунок 8 – Кількість медзакладів за районами Києва



Рисунок 9 – Графік розподілу медичних закладів по районах Києва

Бачимо, що найбільше медзакладів у Шевченківському, Солом’янському та Оболонському районах Києва.

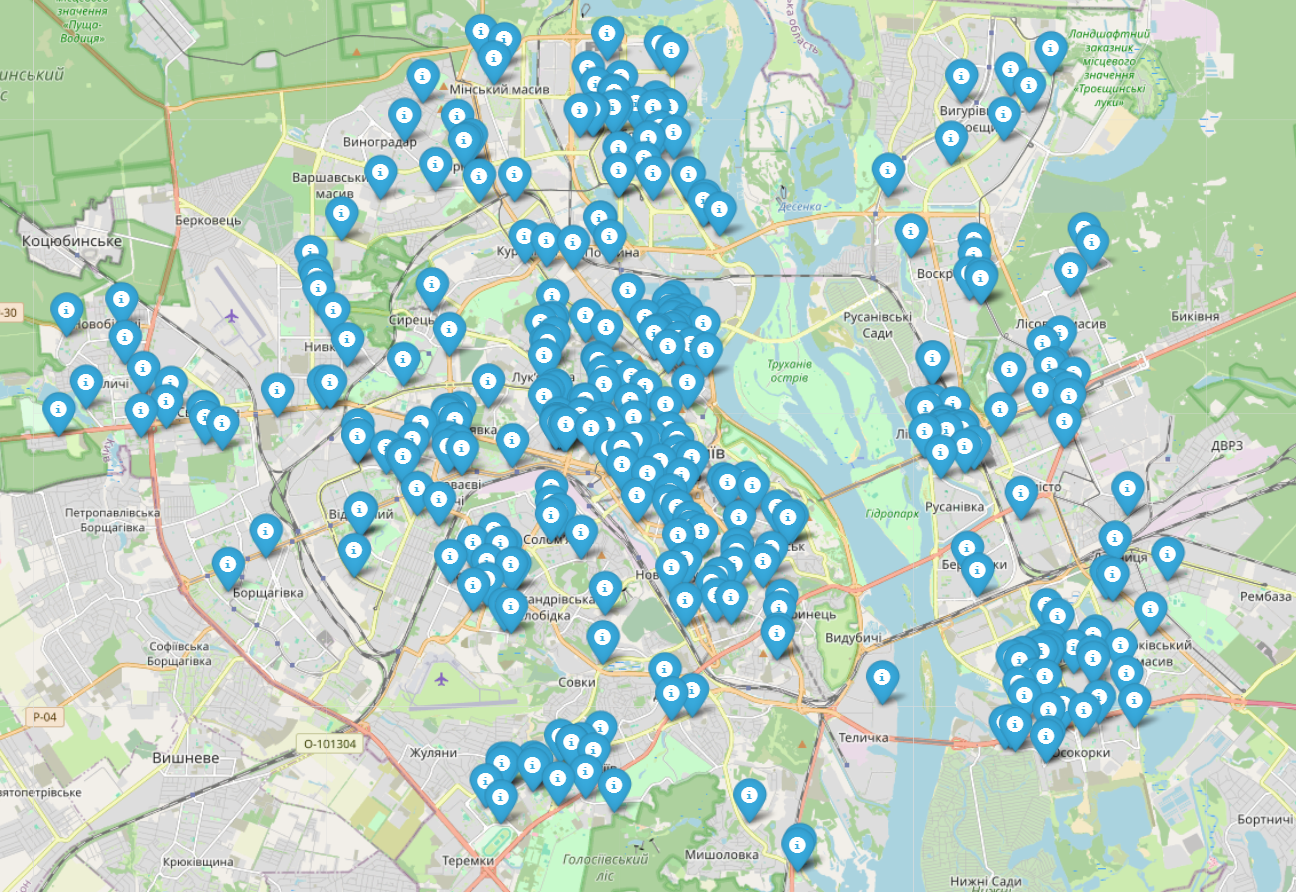


Рисунок 10 – Медзаклади на карті Києва

1. На рисунках 11-12 зображено результати роботи алгоритму №2:

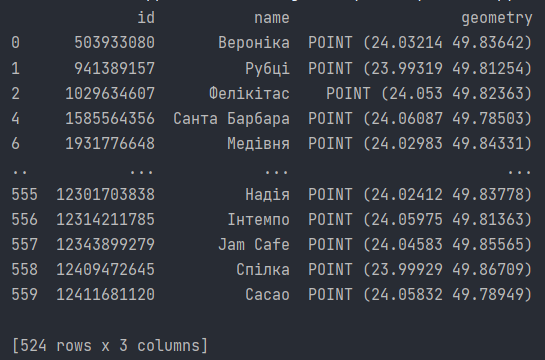


Рисунок 11 – Дані про кав’ярні Львова

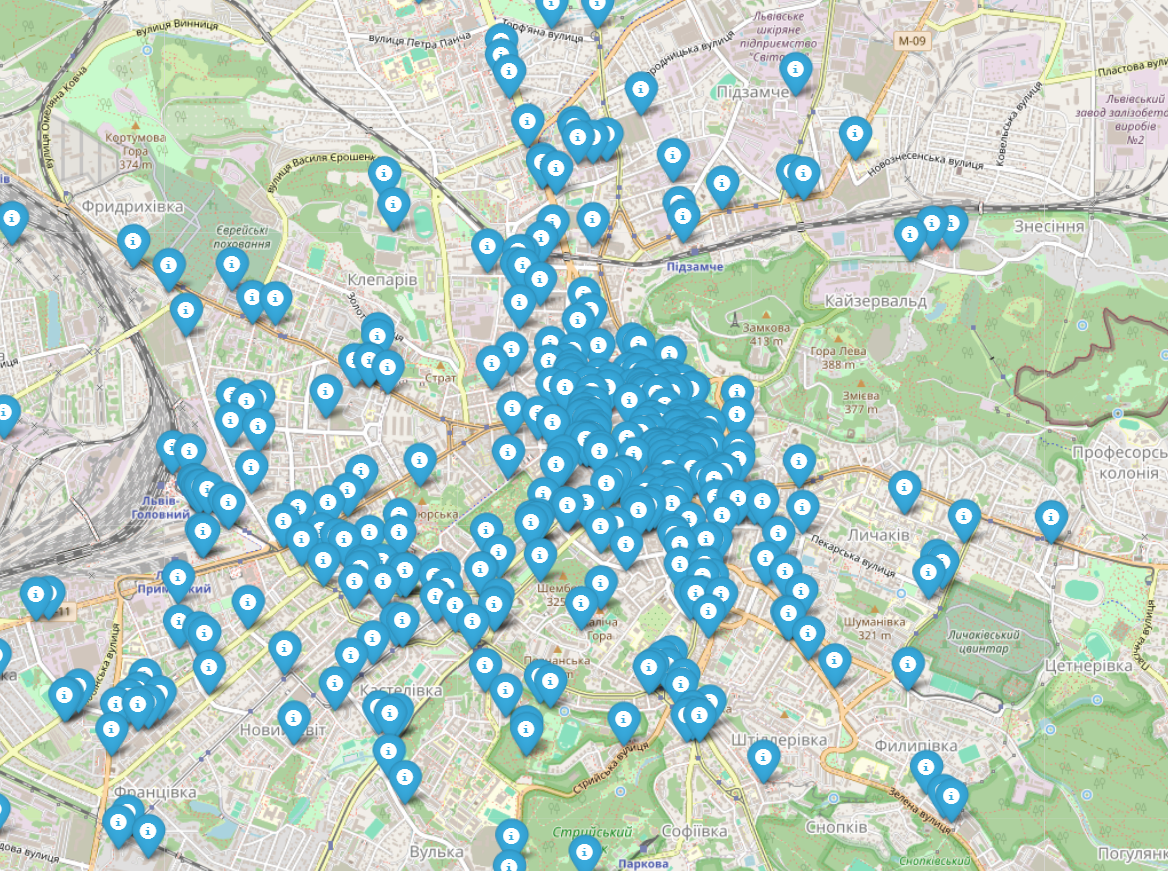


Рисунок 12 – Кав’ярні на карті Львова

Бачимо, що більшість кав’ярень зосереджена ближче до центру міста.

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

Програмний код послідовно реалізує алгоритми на рис.1-2 та спрямований на отримання результатів, поданих на рис.4-12.

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

При цьому використано можливості Python бібліотек: geopy, matplotlib, time, osmnx, time, folium.

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

hospitals\_in\_kyiv.py

*from* geopy.geocoders *import* Nominatim  
*import* osmnx *as* ox  
*import* time  
*import* matplotlib.pyplot *as* plt  
*import* folium  
*from* geopy.exc *import* GeocoderTimedOut  
  
*# Початкові параметри*city = "Kyiv, Ukraine"  
tags = {"amenity": ["hospital", "clinic"]} *# Медичні заклади  
  
# Завантаження даних*data = ox.features\_from\_place(city, tags)  
data = data.reset\_index()  
  
*# Вибір основної інформації*selected\_columns = ['id', 'name', 'amenity', 'geometry']  
data\_cleaned = data[selected\_columns]  
  
*# Додавання координат із геометрії*data\_cleaned['latitude'] = data\_cleaned['geometry'].apply(*lambda* geom: geom.y *if* geom.geom\_type == 'Point' *else None*)  
data\_cleaned['longitude'] = data\_cleaned['geometry'].apply(*lambda* geom: geom.x *if* geom.geom\_type == 'Point' *else None*)  
  
*# Перевірка записів із відсутніми або некоректними координатами та іменами і їх видалення*invalid\_coords = data\_cleaned[(data\_cleaned['latitude'].isna()) | (data\_cleaned['longitude'].isna())]  
data\_cleaned = data\_cleaned.dropna(subset=['latitude', 'longitude']).reset\_index(drop=*True*)  
data\_cleaned = data\_cleaned.loc[~data\_cleaned['name'].isna()]  
  
print(data\_cleaned[['name', 'amenity', 'geometry']])  
  
  
*# Підрахунок кількості закладів за типом*counts = data\_cleaned['amenity'].value\_counts()  
  
print("Розподіл медичних закладів за типами:")  
print(counts)  
  
*# Побудова графіка*counts.plot(kind='bar', color='skyblue')  
plt.title("Розподіл медичних закладів за типами у Києві")  
plt.xlabel("Тип закладу")  
plt.ylabel("Кількість")  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()  
  
  
*# Підрахунок кількості закладів за районами Києва*locator = Nominatim(user\_agent="medical\_map", timeout=5)  
  
  
*# Функція для визначення району  
def* get\_borough(lat, lon):  
 *try*:  
 location\_ = locator.reverse((lat, lon), exactly\_one=*True*)  
 *if* location\_:  
 address = location\_.raw['address']  
 *return* address['borough']  
 *return* "Unknown"  
 *except* GeocoderTimedOut:  
 *return* "Timeout"  
  
  
*# Додавання районів до кожного запису*boroughs = []  
*for* \_, row *in* data\_cleaned.iterrows():  
 borough = get\_borough(row['latitude'], row['longitude'])  
 boroughs.append(borough)  
 time.sleep(0.5) *# Щоб не перевищити ліміт запитів*data\_cleaned['borough'] = boroughs  
  
*# Перевірка результату*print(data\_cleaned[['name', 'latitude', 'longitude', 'borough']].head())  
  
*# Групування за районами та підрахунок кількості*borough\_counts = data\_cleaned['borough'].value\_counts()  
  
*# Вивід результатів*print("Кількість медичних закладів за районами Києва:")  
print(borough\_counts)  
  
*# Побудова графіка*borough\_counts.plot(kind='bar', color='green')  
plt.title("Кількість медичних закладів за районами Києва")  
plt.xlabel("Район")  
plt.ylabel("Кількість")  
plt.xticks(rotation=90)  
plt.tight\_layout()  
plt.show()  
  
  
*# Відображення даних на карті*location = locator.geocode(city)  
center\_coords = [location.latitude, location.longitude]  
  
*# Створення карти*map\_kyiv = folium.Map(location=center\_coords, zoom\_start=12)  
  
*# Додавання точок  
for* \_, row *in* data\_cleaned.iterrows():  
 coords = (row['latitude'], row['longitude'])  
 name = f"{row['name']} ({row['amenity']})"  
 folium.Marker(  
 location=coords,  
 popup=name,  
 icon=folium.Icon(color="blue", icon="info-sign")  
 ).add\_to(map\_kyiv)  
  
*# Збереження карти та її відкриття у браузері*map\_kyiv.save("kyiv\_medical\_facilities.html")  
map\_kyiv.show\_in\_browser()

cafes\_in\_lviv.py

*from* geopy.geocoders *import* Nominatim  
*import* osmnx *as* ox  
*import* folium  
  
*# Початкові параметри*city = "Lviv, Ukraine"  
tags = {"amenity": ["cafe"]} *# Кафе  
  
# Завантаження даних*data = ox.features\_from\_place(city, tags)  
data = data.reset\_index()  
  
*# Вибір основної інформації*selected\_columns = ['id', 'name', 'geometry']  
data\_cleaned = data[selected\_columns]  
*# Очищення даних*data\_cleaned = data\_cleaned.loc[data\_cleaned['geometry'].geom\_type == 'Point']  
data\_cleaned = data\_cleaned.loc[~data\_cleaned['name'].isna()]  
  
print(data\_cleaned)  
  
  
*# Відображення даних на карті*locator = Nominatim(user\_agent="cafes\_map")  
location = locator.geocode(city)  
center\_coords = [location.latitude, location.longitude]  
  
*# Створення карти*map\_lviv = folium.Map(location=center\_coords, zoom\_start=14)  
  
*# Додавання точок  
for* \_, row *in* data\_cleaned.iterrows():  
 coords = (row['geometry'].y, row['geometry'].x)  
 folium.Marker(  
 location=coords,  
 popup=row['name'],  
 icon=folium.Icon(color="blue", icon="info-sign")  
 ).add\_to(map\_lviv)  
  
*# Збереження карти та її відкриття у браузері*map\_lviv.save("lviv\_cafes.html")  
map\_lviv.show\_in\_browser()

## **IV. Висновки.**

У ході виконання лабораторної роботи було досліджено, виявлено та узагальнено особливості впровадження геоінформаційних технологій в процесах Data Science. Було отримано геоінформацію про медичні заклади Києва, виконано аналіз та позначено їх на карті міста, було отримано геоінформацію про кав’ярні Львова та позначено їх на карті.

Виконав: студент Трикош І. В.