Лабораторна робота № 1. Інтегрований аналіз даних в Data Science: практичне застосування та візуалізація

Мета: сформулювати практичні навички аналізу та візуалізації даних в контексті Data Science. Розвинути компетенції в обробці даних, їх статистичному аналізі, моделюванні, а також візуалізації результатів аналізу для ефективної інтерпретації.

Теоретичні відомості

В області Data Science інтегрований аналіз даних відіграє ключову роль, об'єднуючи статистичний аналіз, машинне навчання, візуалізацію та обробку даних для отримання цінної інформації з великих та різноманітних наборів даних. Аналіз даних охоплює широкий спектр методів та процедур, які дозволяють перетворити сиру інформацію у корисні знання та інсайти.

Обробка даних передбачає очищення, нормалізацію, трансформацію та агрегування даних. Очищення даних включає в себе видалення або корекцію неповних, некоректних або нерелевантних частин даних. Нормалізація даних – це процес приведення різноманітних форматів даних до єдиного стандарту, що спрощує аналіз.

Статистичний аналіз забезпечує математичне осмислення даних, дозволяючи визначити тенденції, залежності та закономірності. Він включає в себе розрахунок основних статистичних показників, таких як середнє значення, медіана, мода, стандартне відхилення, а також складніші методи, такі як кореляційний аналіз та регресійний аналіз.

Машинне навчання використовується для створення моделей, які можуть робити прогнози або класифікувати дані на основі історичних даних. Воно охоплює різні типи навчання, включаючи навчання з учителем, без учителя та підкріплювальне навчання.

Візуалізація даних – це ефективний спосіб комунікації інформації, який дозволяє швидко ідентифікувати нові тренди та залежності. Вона включає в себе створення різних типів графіків, таких як лінійні діаграми, гістограми, точкові діаграми, коробчасті діаграми, теплові карти та інші.

Комплексний підхід до аналізу даних у Data Science вимагає знань та навичок у всіх цих областях, дозволяючи проводити глибокий та всеохоплюючий аналіз даних.

Приклад виконання роботи

Завдання

1. Створіть DataFrame зі стовпцями «City», «Latitude» та «Longitude». Згенеруйте дані для 10 різних міст з випадковими координатами.
2. Додайте стовпець «Distance_to_Capital», що показує відстань кожного міста до столиці вашої країни.
3. Створіть стовпець «Region», який категоризує міста на «North», «South», «East», «West» на основі їхніх координат.
4. Використовуючи бібліотеку для візуалізації геоданих (наприклад, Geopandas), створіть хороплетну карту, яка відображає «City» залежно від «Region».
5. Додайте стовпець «Population» з випадковими значеннями. Проаналізуйте кореляцію між «Population» та «Distance_to_Capital».

Контрольне запитання

Вкажіть, яка кількість міст визначена як «East» у стовпці «Region»?
Відповідь: _____

Хід роботи

1. Імпорт бібліотек, що будуть використані в ході роботи

```
import pandas as pd
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt
from geopy.distance import great_circle
```

2. Створення DataFrame зі стовпцями «City», «Latitude» та «Longitude»

```
# Завдання №1. Створення Dataframe
np.random
cities = [f'City_{i}' for i in range(1, 11)]
latitudes = np.random.uniform(-90, 90, size=10)
longitudes = np.random.uniform(-180, 180, size=10)

df = pd.DataFrame({'City': cities, 'Latitude': latitudes, 'Longitude':
longitudes})
df.head(10)
```

	City	Latitude	Longitude
0	City_1	77.582328	-26.348098
1	City_2	13.785159	82.280663
2	City_3	61.136138	92.280741
3	City_4	22.192580	-36.859938
4	City_5	-31.586851	153.072039
5	City_6	41.042096	-106.736284
6	City_7	4.092593	-177.119024
7	City_8	42.626064	153.486255
8	City_9	-60.226899	-73.975844
9	City_10	33.670493	-119.897426

Рисунок 1.1. Результат виконання завдання №1

3. Реалізація стовпця «Distance_to_Capital»

```
# Завдання №2. Обчислення відстаней до столиці

capital_coords = (0, 0) # столиця знаходиться в координатах (0;0)
def calculate_distance(lat, lon):
    return great_circle(capital_coords, (lat, lon)).kilometers

df['Distance_to_Capital'] = df.apply(lambda x:
calculate_distance(x['Latitude'], x['Longitude']), axis=1)
df.head(10)
```

	City	Latitude	Longitude	Distance_to_Capital
0	City_1	-1.121451	175.901816	19542.691359
1	City_2	21.592029	28.708279	3931.877434
2	City_3	3.550000	-43.149178	4811.003766
3	City_4	-61.777549	18.341359	7041.834219
4	City_5	-86.656284	88.320395	9996.665953
5	City_6	-77.396014	60.923842	9330.674715
6	City_7	-2.457880	-84.628959	9410.875265
7	City_8	19.139303	-156.119460	16651.619365
8	City_9	12.393259	-46.769689	5338.723290
9	City_10	-32.874766	46.698303	6096.791795

Рисунок 1.2. Результат виконання завдання №2

4. Реалізація стовпця «Region»

```
# Завдання №3. Категоризація міст
def categorize_region(lat, lon):
    if lat >= 0:
        return 'North' if lon >= 0 else 'West'
    else:
        return 'South' if lon >= 0 else 'East'

df['Region'] = df.apply(lambda x: categorize_region(x['Latitude'],
x['Longitude']), axis=1)
df.head(10)
```

	City	Latitude	Longitude	Distance_to_Capital	Region
0	City_1	-1.121451	175.901816	19542.691359	South
1	City_2	21.592029	28.708279	3931.877434	North
2	City_3	3.550000	-43.149178	4811.003766	West
3	City_4	-61.777549	18.341359	7041.834219	South
4	City_5	-86.656284	88.320395	9996.665953	South
5	City_6	-77.396014	60.923842	9330.674715	South
6	City_7	-2.457880	-84.628959	9410.875265	East
7	City_8	19.139303	-156.119460	16651.619365	West
8	City_9	12.393259	-46.769689	5338.723290	West
9	City_10	-32.874766	46.698303	6096.791795	South

Рисунок 1.3. Результат виконання завдання №3

5. Візуалізація геоданих за допомогою «Geopandas»

```
# Завдання №4. Хороплетена карта

gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.Longitude,
df.Latitude))

world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
fig, ax = plt.subplots(figsize=(20, 6))
world.plot(ax=ax, color='lightgrey', edgecolor='black')
gdf.plot(ax=ax, color='red', marker='o', markersize=100, label='Cities')
plt.title('Географічне Розташування Міст', fontsize=20, fontweight='bold')
plt.xlabel('Довгота', fontsize=15)
plt.ylabel('Широта', fontsize=15)
plt.legend(fontsize=12)
plt.show()
```

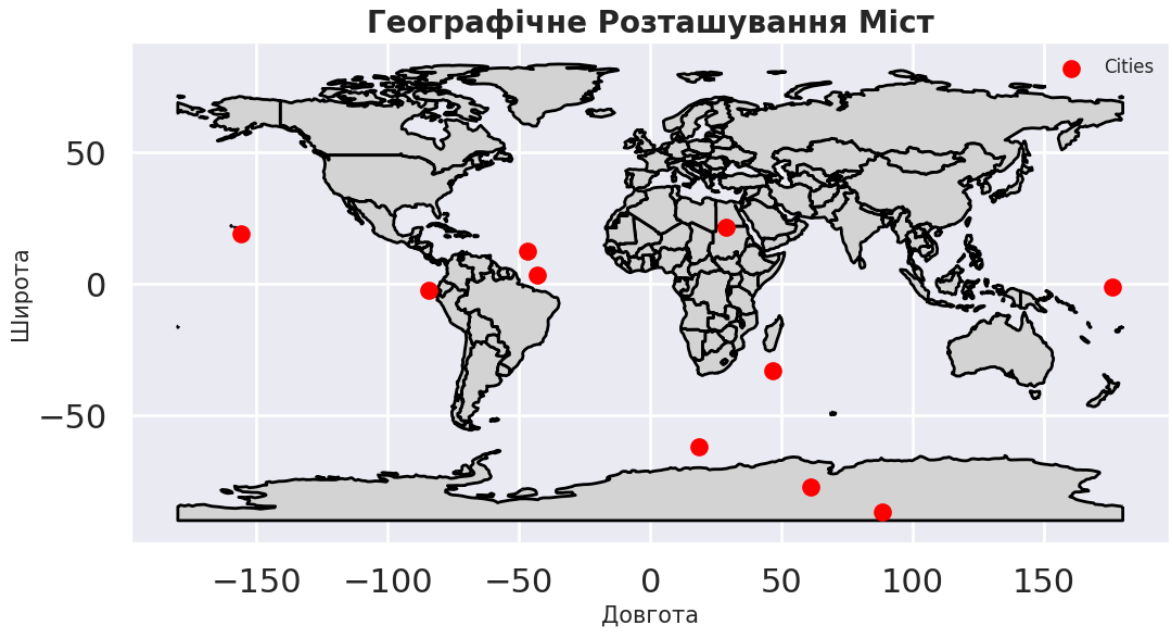

Рисунок 1.4. Результат виконання завдання №4

6. Аналіз кореляції між новим стовпцем «Population» та «Distance_to_Capital»

```
# Завдання №5. Кореляція між 'Population' та 'Distance_to_Capital'
df['Population'] = np.random.randint(1000, 100000, size=10)
df.head(10)
correlation = df[['Population', 'Distance_to_Capital']].corr()
print("Кореляція між населенням та відстанню до столиці:")
print(correlation)
```

	City	Latitude	Longitude	Distance_to_Capital	Region	Population
0	City_1	-1.121451	175.901816	19542.691359	South	92462
1	City_2	21.592029	28.708279	3931.877434	North	39553
2	City_3	3.550000	-43.149178	4811.003766	West	20686
3	City_4	-61.777549	18.341359	7041.834219	South	36056
4	City_5	-86.656284	88.320395	9996.665953	South	79896
5	City_6	-77.396014	60.923842	9330.674715	South	1690
6	City_7	-2.457880	-84.628959	9410.875265	East	7078
7	City_8	19.139303	-156.119460	16651.619365	West	26900
8	City_9	12.393259	-46.769689	5338.723290	West	58961
9	City_10	-32.874766	46.698303	6096.791795	South	60574

Рисунок 1.5. Результат виконання завдання №5

```
Кореляція між населенням та відстанню до столиці:
                    Population  Distance_to_Capital
Population             1.00000              0.30226
Distance_to_Capital    0.30226              1.00000
```

Рисунок 1.6. Результат виконання завдання №5

Контрольне запитання

Вкажіть, яка кількість міст визначена як «East» у стовпці «Region»?

Відповідь: **1**

Висновки: …

Завдання для виконання лабораторної роботи

Варіант 1.

1. Імпортуйте набір даних у DataFrame з файлу CSV. Файл повинен містити дані з трьома стовпцями: «Name» (string), «Age» (int), «City» (string). Кількість рядків повинна бути більшою за 30.
2. Видаліть стовпець «City» з імпортованого DataFrame.
3. Перейменуйте стовпець «Name» на «FirstName» та «Age» на «PersonAge».
4. Відфільтруйте дані, щоб в DataFrame залишились тільки рядки, де значення в «PersonAge» більше 30 років.
5. Виведіть 2, 6, 12, 24 рядки оновленого DataFrame.

Контрольне запитання

Вкажіть значення «PersonAge» у 12 рядку фінального DataFrame.

Відповідь: _____

Варіант 2.

1. Створіть DataFrame з двома стовпцями: «Temperature» (температура) та «Humidity» (вологість). Згенеруйте 100 випадкових значень для кожного стовпця, де «Temperature» в межах від -10 до 30 градусів, а «Humidity» – від 30% до 90%.
2. На основі значень у стовпці «Temperature» створіть новий стовпець «Temperature_Category», який класифікує температуру як «Low» (низька) для значень < 0, «Moderate» (помірна) для значень від 0 до 20, та «High» (висока) для значень > 20.
3. Створіть стовпець «Comfortable», де значення буде True, якщо «Temperature» знаходиться в межах від 18 до 24 градусів та «Humidity» від 40% до 60%, інакше False.
4. Додайте стовпець «Heat_Index», який розраховується за формулою:
   $Heat\_Index = \frac{Temperature \cdot Humidity}{100}$
5. Виведіть описову статистику (середнє, мінімальне, максимальне значення) для стовпців «Temperature», «Humidity» та «Heat_Index».

Контрольне запитання

Вкажіть кількість значень «True» в стовпці «Comfortable».

Відповідь: _____

Варіант 3.

1. Створіть DataFrame з трьома стовпцями: «Product», «Sales», та «Region». «Product» містить назви трьох продуктів, «Sales» – кількість продажів (випадкові числа), «Region» – один з трьох регіонів продажу. Згенеруйте 50 випадкових записів.
2. Створіть гістограму для візуалізації розподілу продажів («Sales»).

3. Візуалізуйте продажі («Sales») для кожного продукту («Product») за допомогою коробчастої діаграми (boxplot).
4. Створіть точкову діаграму (scatter plot), щоб показати залежність між «Sales» та «Region».
5. Використовуйте стовпчикову діаграму для порівняння загальних продажів по кожному регіону («Region»).

Контрольне запитання

Вкажіть регіон з найбільшою кількістю загальних продажів.
Відповідь: _____

Варіант 4.

1. Створіть DataFrame з двома стовпцями: «Age» та «Income». Згенеруйте 100 випадкових значень для кожного стовпця, де «Age» випадково розподілене від 18 до 65 років, а «Income» – від 30.000 до 100.000.
2. Виведіть основні статистичні показники (середнє, медіана, стандартне відхилення) для «Age» та «Income».
3. Обчисліть та візуалізуйте кореляцію між «Age» та «Income».
4. Створіть новий стовпець «Log_Income», який є логарифмом значень «Income».
5. Виконайте нормалізацію стовпця «Income» за допомогою Min-Max scaling і додайте це як новий стовпець «Normalized_Income».

Контрольне запитання

Вкажіть показник «середнє» для «Income».
Відповідь: _____

Варіант 5.

1. Створіть DataFrame зі стовпцем «Date», який містить щоденні дати з початку поточного року до сьогодні. Додайте стовпець «Sales», що містить випадкові значення продажів.
2. Згрупуйте дані за місяцями та обчисліть загальні продажі в кожному місяці.
3. Якщо у «Sales» є пропущені значення, заповніть їх середнім значенням продажів.
4. Створіть стовпець «Previous_Day_Sales», який містить значення продажів попереднього дня. Для першого запису використовуйте значення 0.
5. Додайте стовпець «Cumulative_Sales», який показує кумулятивну (загальну) суму продажів з початку року.

Контрольне запитання

Вкажіть показник кумулятивної (загальної) суми продажів з початку року.
Відповідь: _____

Варіант 6.

1. Створіть два DataFrame: один зі стовпцями «ID», «Name», «Age» і другий зі стовпцями «ID», «Occupation». Згенеруйте 50 випадкових записів для кожного DataFrame і виконайте об'єднання (merge) даних за стовпцем «ID».
2. У результуючому DataFrame після об'єднання, створіть новий стовпець «Age_Group», який класифікує «Age» на категорії «Young», «Adult», «Senior».
3. Застосуйте функції для обробки текстових даних, щоб замінити символи «r» на «g» у стовпці «Name».
4. Використовуйте метод apply для створення нового стовпця «Length_Name», який містить кількість символів у кожному імені у стовпці «Name».
5. Створіть візуалізацію, що поєднує гістограму розподілу «Age» та кругову діаграму розподілу «Occupation».

Контрольне запитання

Вкажіть кількість рядків, що мають категорію «Senior» у стовпці «Age_Group».
Відповідь: _____

Варіант 7.

1. Створіть DataFrame зі стовпцями «Month», «Product_A_Sales», «Product_B_Sales». Згенеруйте 12 рядків, що представляють місяці року, та випадкові значення продажів для двох продуктів.
2. Створіть лінійний графік для порівняння місячних продажів «Product_A_Sales» та «Product_B_Sales».
3. Створіть гістограму для візуалізації розподілу продажів «Product_A_Sales».
4. Використовуйте скрипкову діаграму для візуалізації розподілу продажів обох продуктів.
5. Створіть теплову карту (heatmap), щоб візуалізувати кореляцію між «Product_A_Sales» та «Product_B_Sales» у різні місяці.

Контрольне запитання

Вкажіть показник кореляції «Product_A_Sales» та «Product_B_Sales» у лютому.
Відповідь: _____

Варіант 8.

1. Створіть DataFrame, який містить стовпець «Timestamp» з випадковими датами та часом протягом останнього року. Додайте стовпець «Sales», що містить випадкові значення продажів.
2. Додайте стовпець «Sales_Change», який показує відсоткову зміну продажів порівняно з попереднім днем.
3. Згрупуйте дані за місяцями та розрахуйте середні продажі в кожному місяці.
4. Створіть графік, який відображає середні місячні продажі.

5. Визначте 5 днів із найвищими продажами та виведіть відповідні дати та значення продажів.

Контрольне запитання

Вкажіть дату дня з найвищими продажами.
Відповідь: _____

Варіант 9.

1. Створіть DataFrame зі стовпцями «Player», «Points», «Assists», «Rebounds». Згенеруйте випадкові спортивні статистики для 30 гравців. Виведіть загальну кількість «Points», «Assists» і «Rebounds» для всіх гравців.
2. Виберіть та виведіть дані гравців, які набрали більше 20 очок («Points») та 5 асистів («Assists»).
3. Відсортуйте DataFrame за «Points» у спадаючому порядку, а потім за «Assists» у зростаючому.
4. Додайте стовпець «Team», який містить назви команд (наприклад, TeamA, TeamB, TeamC). Розподіліть гравців по цих трьох командах випадковим чином.
5. Визначте середню кількість «Points» та «Rebounds» для кожної команди.

Контрольне запитання

Вкажіть кількість гравців, що були розподілені до першої команди.
Відповідь: _____

Варіант 10.

1. Створіть DataFrame зі стовпцями «Name» та «Review». «Name» містить імена осіб, а «Review» – короткі текстові відгуки. Згенеруйте 50 випадкових записів.
2. Переведіть усі відгуки («Review») до нижнього регістру.
3. Використовуйте регулярні вирази для видалення всіх цифр із текстових відгуків.
4. Додайте стовпець «Word_Count», який показує кількість слів у кожному відгуку.
5. Створіть стовпець «Sentiment», який категоризує кожен відгук як «Positive», «Neutral», або «Negative» на основі випадково згенерованих значень.

Контрольне запитання

Вкажіть кількість відгуків, що були категоризовані як «Neutral».
Відповідь: _____

Варіант 11.

1. Створіть DataFrame зі стовпцями «Date» (дати за 2023 рік) та «Stock_Price». Згенеруйте випадкові значення для «Stock_Price».
2. Обчисліть та додайте стовпець з 7-денним рухомим середнім для «Stock_Price».

3. Виконайте сезонну декомпозицію для «Stock_Price» та візуалізуйте результат.
4. Обчисліть та візуалізуйте лагову кореляцію (кореляції між значеннями в часовому ряді) для «Stock_Price».
5. Використовуйте просту лінійну регресію для прогнозування «Stock_Price» на наступні 30 днів на основі існуючих даних.

Контрольне запитання

Вкажіть результат прогнозування «Stock_Price» на 30 день.

Відповідь: _____

Варіант 12.

1. Створіть DataFrame зі стовпцями «Date», «Company», «Revenue» та «Expenses». Згенеруйте дані для трьох компаній за останній рік з випадковими значеннями доходів та витрат.
2. Додайте стовпець «Profit», що розраховується як різниця між «Revenue» та «Expenses».
3. Групуйте дані за «Company» та обчисліть загальний прибуток для кожної компанії.
4. Визначте, в яку дату кожна компанія мала максимальний та мінімальний прибуток.
5. Створіть графіки, що показують місячні зміни «Revenue», «Expenses» та «Profit» для кожної компанії.

Контрольне запитання

Вкажіть значення мінімального прибутку компанії за весь період.

Відповідь: _____

Варіант 13.

1. Створіть DataFrame зі стовпцями «Person_ID», «Age», «Income», «Social_Media_Hours». Згенеруйте дані для 100 осіб з випадковими значеннями для кожного стовпця.
2. Категоризуйте осіб за віком: «Youth» (до 18 років), «Adult» (від 18 до 60 років) та «Senior» (понад 60 років).
3. Виведіть кореляційну матрицю для всіх числових стовпців.
4. Створіть гістограму, що показує розподіл кількості годин, проведених в соціальних мережах.
5. Обчисліть та візуалізуйте середній дохід у кожній віковій групі.

Контрольне запитання

Вкажіть середній дохід у віковій групі «Youth».

Відповідь: _____

Варіант 14.

1. Створіть DataFrame, який відображає щотижневі продажі продукту протягом року. Дані повинні включати стовпці «Week» та «Sales».
2. Проаналізуйте часовий ряд, виявіть основні тренди та сезонні коливання в даних.
3. Обчисліть 4-тижневу рухому середню продажів та додайте її як новий стовпець «Mean_Week» у DataFrame.
4. Використовуючи будь-яку методику прогнозування часових рядів, створіть прогноз продажів на наступні 3 місяці і візуалізуйте його разом із історичними даними.
5. Розрахуйте та проаналізуйте відхилення фактичних продажів від прогнозованих за останній місяць.

Контрольне запитання

Вкажіть тиждень, в якому було виконано найбільше продажів.

Відповідь: _____

Варіант 15.

1. Створіть DataFrame, який містить стовпці «Student_ID», «Subject», «Score». Згенеруйте дані для оцінок студентів з різних предметах.
2. Проведіть аналіз розподілу оцінок у кожному предметі. Визначте предмети з найвищими та найнижчими середніми оцінками.
3. Створіть візуалізації, що відображають розподіл оцінок у кожному предметі.
4. Розрахуйте середню оцінку кожного студента з усіх предметів.
5. Визначте студентів з найбільшими відхиленнями в оцінках між предметами.

Контрольне запитання

Вкажіть «Student_ID» студента, що має найбільше відхилення в оцінках.

Відповідь: _____

Варіант 16.

1. Створіть DataFrame зі стовпцями «Patient_ID», «Age», «Blood_Pressure», «Cholesterol_Level». Згенеруйте дані для 100 пацієнтів з випадковими значеннями для кожного стовпця.
2. Використовуючи дані про кров'яний тиск («Blood_Pressure») та рівень холестерину («Cholesterol_Level»), ідентифікуйте пацієнтів з потенційно високим ризиком серцевих захворювань. Створіть новий стовпець («Health_Hazards»).
3. Створіть графіки, які відображають розподіл «Blood_Pressure» та «Cholesterol_Level» серед пацієнтів.
4. Проаналізуйте середній «Blood_Pressure» та «Cholesterol_Level» у різних вікових групах.
5. Обчисліть кореляцію між «Age», «Blood_Pressure» та «Cholesterol_Level» та інтерпретуйте отримані результати.

Контрольне запитання

Вкажіть найбільший показник у стовпці «Health_Hazards».
Відповідь: _____

Варіант 17.

1. Створіть DataFrame зі стовпцями «Company_ID», «Quarter», «Revenue», «Expenses». Згенеруйте дані для кількох компаній за останні два роки.
2. Визначте компанії з найвищим та найнижчим квартальним приростом доходів.
3. Створіть лінійні діаграми, що відображають динаміку доходів і витрат для обраних компаній.
4. Розрахуйте маржу прибутку для кожної компанії та аналізуйте її зміни протягом часу.
5. Виконайте дослідження кореляції між «Revenue» та «Expenses» у компаніях.

Контрольне запитання

Вкажіть назву компанії з найвищим квартальним приростом доходів.
Відповідь: _____

Варіант 18.

1. Створіть DataFrame зі стовпцями «Incident_ID», «Date», «Type», «Location». Згенеруйте дані про різні безпекові інциденти, що сталися протягом останнього року.
2. Визначте, які типи інцидентів трапляються найчастіше, використовуючи групування та підрахунок.
3. Створіть графік, що показує кількість інцидентів у різних місцях.
4. Визначте, в які періоди року інциденти трапляються частіше.
5. Виконайте ручну заміну «Location» для 2, 6, 12 на NaN (Null).

Контрольне запитання

Вкажіть період року з найменшою кількістю інцидентів.
Відповідь: _____

Варіант 19.

1. Створіть DataFrame зі стовпцями «Observation_ID», «Date», «Time», «Location», «Description». Згенеруйте дані про заявлені спостереження невпізнаних літаючих об'єктів (НЛО) за останні декілька років.
2. Вивчіть розподіл спостережень НЛО за географічними регіонами та часом (в період з 00:00 до 06:00 та в період з 06:01 до 23:59).
3. Створіть графіки, які відображають розподіл спостережень за роками та місцезнаходженням.

4. Проаналізуйте описи спостережень, щоб виявити найчастіші слова або фрази, які використовуються свідками.
5. На основі описів та даних, оцініть потенційну надійність спостережень. Визначте, які з них можуть бути високо вірогідними та які – можливими помилковими повідомленнями.

Контрольне запитання

Вкажіть локацію де кількість заявлених спостережень є максимальною.
Відповідь: _____

Варіант 20.

1. Створіть DataFrame зі стовпцями «Car_ID», «Model», «Year», «Battery_Capacity», «Range». Згенеруйте дані про різні моделі автомобілів Tesla, включаючи інформацію про рік випуску, ємність батареї та пробіг.
2. Дослідіть, як ємність батареї впливає на загальний пробіг автомобіля.
3. Створіть лінійні діаграми або скатерплоти, що показують еволюцію ємності батареї та пробігу автомобілів Tesla за роками.
4. Порівняйте різні моделі автомобілів Tesla за ключовими характеристиками, включаючи ємність батареї та пробіг.
5. Розрахуйте та проаналізуйте показники енергоефективності для різних моделей Tesla, враховуючи ємність батареї та пробіг.

Контрольне запитання

Вкажіть модель автомобіля Tesla, яка має найбільший пробіг.
Відповідь: _____

Welcome to Google Colab

https://colab.research.google.com/drive/1NIK8xBm8ZbmzhuqUUgIcwbY3YtAHJjOZ#scrollTo=Je6rmFt5ApbY

Головне про Google Colab: **Google Colab** – це хмарний сервіс від Google, який дозволяє писати, запускати та ділитися кодом на Python за допомогою Jupyter Notebooks. Він надає безкоштовний доступ до обчислювальних ресурсів, що робить його чудовим вибором для запуску скриптів Python, особливо якщо ви не хочете встановлювати Python на своїй локальній машині.

Інструкція з використання Google Colab:

Старт роботи з Google Colab:

1. Перейдіть на https://colab.research.google.com/ і увійдіть за допомогою свого облікового запису Google.

Створіть новий блокнот або використовуйте вже готовий за посиланням https://colab.research.google.com/drive/1NIK8xBm8ZbmzhuqUUgIcwbY3YtAHJjOZ#scrollTo=Je6rmFt5ApbY

1. Натисніть «*File*» у верхньому лівому куті, а потім натисніть «*New notebook*». Відкриється нове вікно Jupyter Notebook у вашому браузері.
2. Натисніть «*Connect*» у верхньому правому куті та досліджуйте розроблений курс.

Перейменування блокноту

1. Натисніть на назву блокноту за замовчуванням «*Untitled0.ipynb*» у верхньому лівому кутку і дайте йому змістовну назву, наприклад, «*Лабораторна робота №1*».

Виконайте завдання лабораторної роботи

1. Використовуйте вже створені комірки блокноту для внесення коду на Python.
2. За необхідності створюйте нові комірки за допомогою комбінації клавіш *Ctrl + M + B* або просто натиснувши «*+ Code*» в проміжку між блоками.
3. Зберігайте позитивний настрій.

За додатковою інформацією щодо Google Colab звертайтеся до документації https://colab-docs.github.io/