МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ЗАТВЕРДЖЕНО
на засіданні кафедри
статистики і економічного прогнозування
Протокол № 2 від 01.09.2025 р.

ПОГОДЖЕНО
Проректор з навчально-методичної роботи



Каріна НЕМАШКАЛО

МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗУ BIG DATA : методичні рекомендації до виконання лабораторних та самостійних робіт

здобувачів вищої освіти спеціальності С1 "Економіка та міжнародні економічні відносини (за спеціалізацією С1.01 Економіка)" освітньої програми "Аналітика даних в економіці" другого (магістерського) рівня

Розробники:
д.е.н., професор підписано КЕП Олена РАЄВНЄВА
д.е.н., професор Костянтин СТРИЖИЧЕНКО
к.е.н., доцент Ольга БРОВКО

Харків
2025

УДК 004.65/.67:330.4
М54

Укладачі: Раєвнєва О. В.
Стрижиченко К. А.
Бровко О. І.

Затверджено на засіданні кафедри статистики і економічного прогнозування.

Протокол № 2 від 01.09.2025 р.

Самостійне електронне текстове мережеве видання

М54 **Методи та програмні засоби аналізу Big Data :** методичні рекомендації до виконання лабораторних та самостійних робіт для здобувачів вищої освіти спеціальності - С1 "Економіка та міжнародні відносини" (за спеціалізацією С1.01.Економіка), освітньої програми "Аналітика даних в економіці" другого (магістерського) рівня: [Електронне видання] / уклад. О. В. Раєвнєва, К. А. Стрижиченко, О. І. Бровко. – Харків : ХНЕУ ім. С. Кузнеця, 2025. – 63 с.

Подано методичні рекомендації до виконання лабораторних та самостійних робіт, метою яких є закріплення й поглиблення знань теоретичного та практичного матеріалу, набуття навичок щодо використання різноманітних методів та підходів в управлінні соціально-економічними системами.

Рекомендовано для студентів спеціальності - С1 "Економіка та міжнародні відносини" (за спеціалізацією С1.01.Економіка), освітньої програми "Аналітика даних в економіці" другого (магістерського) рівня.

УДК 004.65/.67:330.4

© Харківський національний економічний університет імені Семена Кузнеця, 2025

ВСТУП

Сучасне глобальне економічне середовище характеризується високим рівнем динамічності, транскордонності та перманентної мінливості, що супроводжується стрімкими структурними зрушеннями, кризовими явищами та непередбачуваними ринковими коливаннями. За таких умов традиційні підходи до аналізу та прогнозування, що базуються на обмежених історичних вибірках, часто втрачають свою ефективність і не встигають за швидкістю змін. Цифровізація світової економіки та лавиноподібне зростання обсягів інформації перетворюють аналіз Big Data на критично важливий інструмент підтримки прийняття управлінських рішень. Володіння методами інтелектуального аналізу великих даних дозволяє компаніям та державним інституціям оперативно реагувати на виклики в режимі реального часу, мінімізувати ризики, виявляти приховані ринкові можливості та гнучко адаптуватися до трансформацій.

У зв'язку з цим виникає гостра потреба у висококваліфікованих фахівцях, які здатні не лише оперувати класичними економічними показниками, а й ефективно працювати з величезними масивами різнорідних, неструктурованих даних. Вивчення навчальної дисципліни «Методи та програмні засоби аналізу Big Data» належить до циклу обов’язкових дисциплін професійної підготовки здобувачів вищої освіти спеціальності С1 Економіка та міжнародні економічні відносини (за спеціалізацією С1.01 Економіка). Дисципліна направлена на використання сучасного програмного забезпечення та в яких ситуаціях застосовувати програмні засоби для аналізу великих даних в залежності від типу, об’єму і структури даних; встановлення залежності, тренду або тенденції на основі аналізу досліджуваних масивів даних; встановлення зв'язку між досліджуваними ознаками; формулювання коректних висновків та використання методів візуалізації даних.

Об’єктом дисципліни є бізнес–процеси та явища масового характеру.

Предметом дисципліни є програмні продукти, сучасні аналітичні методів та інструменти аналізу даних та обробки бізнес-інформації.

Мета навчальної дисципліни є формування теоретичних знань та практичних навичок моделювання динамічних економічних процесів на підставі використання методів інтелектуального аналізу великих масивів даних.

Методичні рекомендації покликані забезпечити чітку координацію самостійної та аудиторної роботи студентів. Під час виконання

лабораторних робіт та підготовки есе у вигляді презентацій, здобувачі вищої освіти здійснюють глибокий аналіз реальних економічних кейсів, первинну обробку та очищення «сирих» даних, побудову аналітичних і прогностичних моделей, а також інтерактивну візуалізацію отриманих результатів.

Комплексний підхід до виконання завдань дозволить здобувачам опанувати інструментарій Data Science, розвинути критичне та аналітичне мислення, набути практичного досвіду розв'язання прикладних економічних задач і сформувати професійні компетентності, необхідні для успішної конкуренції на сучасному ринку праці.

Змістовий модуль 1. Методологічні засади інтелектуального аналізу Big Data

Тема 1. «Поняття Big Data, принципи та технології управління соціально-економічними процесами та системами»

Лабораторна робота "Обробка таблиць, редагування Big Data та формування баз даних за допомогою мови запитів SQL"

Мета роботи – набуття практичних навичок проектування архітектури сучасних хмарних сховищ великих даних (Data Warehouse), опанування методів пакетного завантаження неструктурованої інформації (Batch Ingestion) та автоматизації процесів очищення і трансформації «сирих» масивів даних (Data Cleaning / ELT). Здобувачі мають навчитися оптимізувати аналітичні SQL-запити до мільйонних датасетів за допомогою технологій секціонування (Partitioning) та кластеризації (Clustering), а також опанувати інструменти глибокого інтелектуального аналізу часових рядів із використанням складних аналітичних віконних функцій у середовищі Google BigQuery Sandbox.

Завдання лабораторної роботи:

розгортання та конфігурація хмарного середовища;
моделювання Staging-рівня сховища великих даних (Bronze Layer);
проєктування оптимізованої архітектури бази даних (Silver Layer);
реалізація ELT-процесу та очищення даних (Data Cleaning);
глибока аналітична обробка даних (DQL);
оптимізація та фінансово-ресурсний аудит запитів.

Методичні рекомендації до виконання лабораторної роботи

Робота з технологіями великих даних (Big Data) вимагає значних обчислювальних потужностей та спеціалізованої інфраструктури, розгортання якої на локальних комп'ютерах користувачів часто є технічно неможливим або нераціональним. Сучасний підхід інженерії даних базується на використанні хмарних платформ (Cloud Computing). **Google Cloud Platform (GCP)** є одним із світових лідерів у сфері аналітики надвеликих масивів інформації завдяки хмарному безсерверному сховищу даних (Serverless Data Warehouse) — **Google BigQuery**.

Крок 1. Підготовка середовища BigQuery Sandbox

1. Перейдіть на сайт Google Cloud Console.

2. Увійдіть під своїм Google-акаунтом.

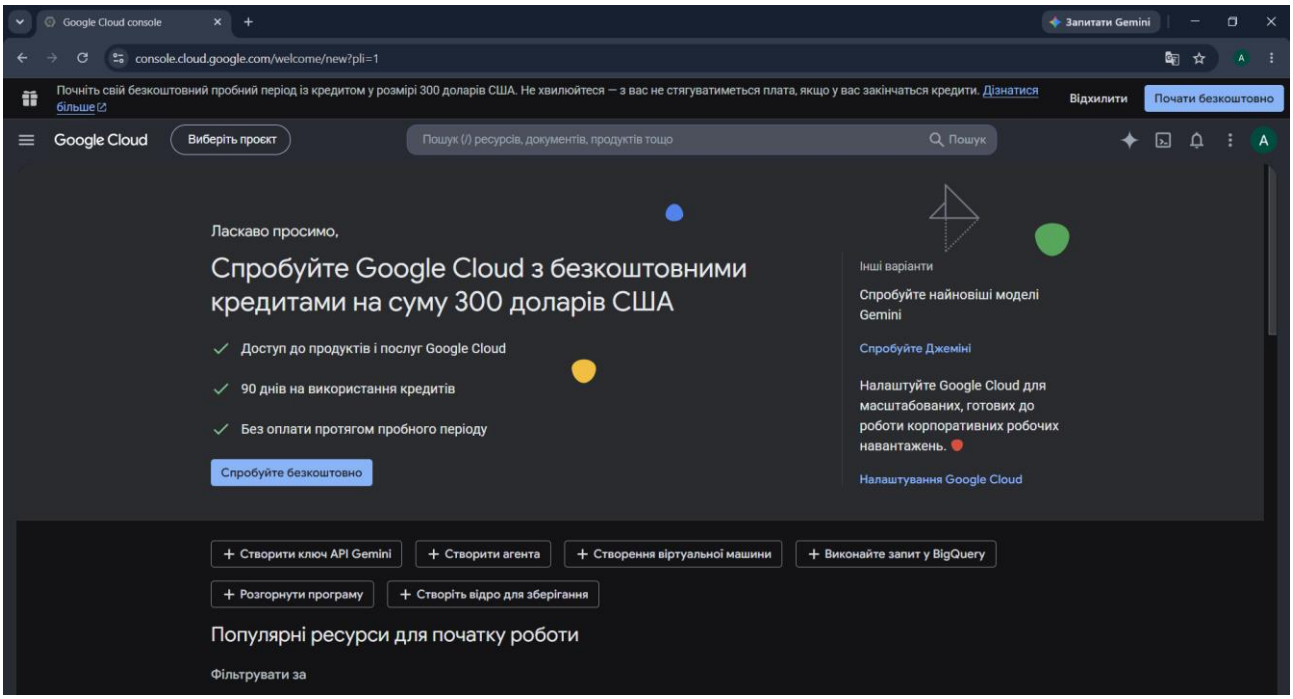


Рис. 1.1. Стартова веб-сторінка консолі керування хмарною платформою Google Cloud Console

3. У верхньому правому куті оберіть «Почати безкоштовно».

4. У верхній панелі натисніть на вибір проєктів та створете новий проєкт (наприклад, bigdata-lab1-yourname).

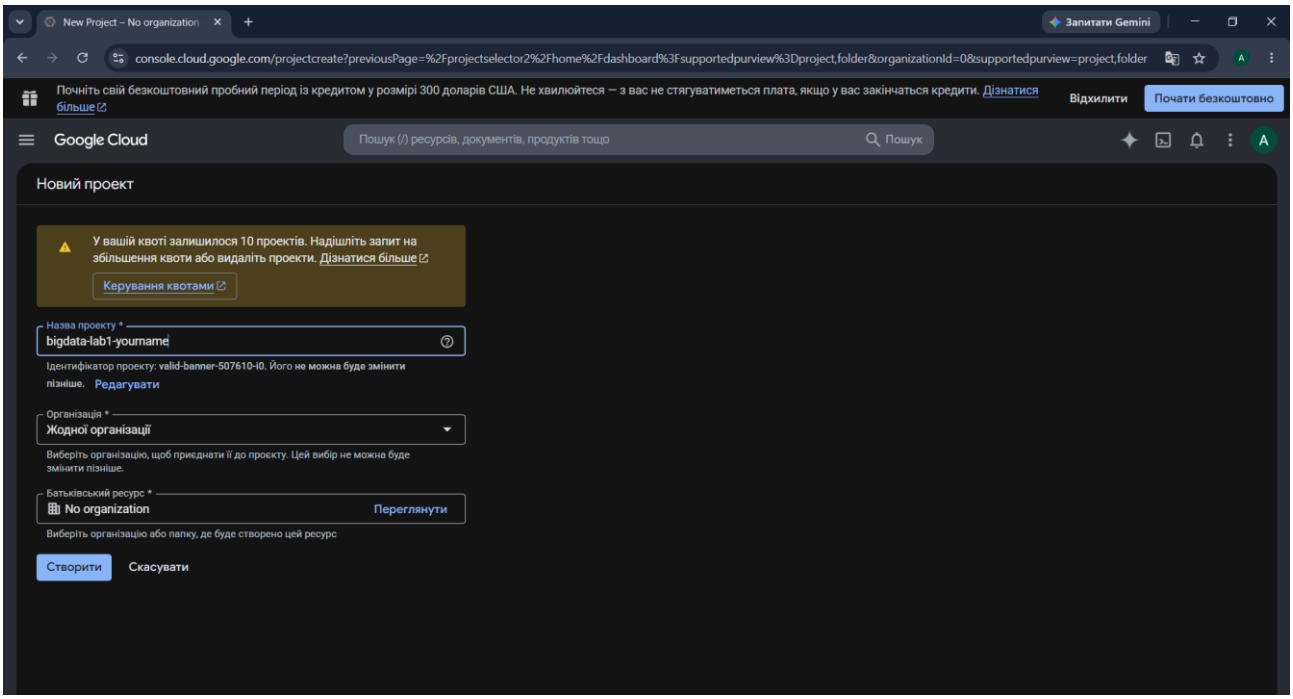


Рис. 1.2. Інтерфейс налаштування та задання параметрів нового хмарного проєкту в Google Cloud

5. У лівому меню знайдіть та відкрийте **BigQuery** (розділ Analytics),

після чого опинитеся у безкоштовній пісочниці (Sandbox), яка дає 10 ГБ безкоштовного зберігання та 1 ТБ безкоштовної обробки запитів на місяць.

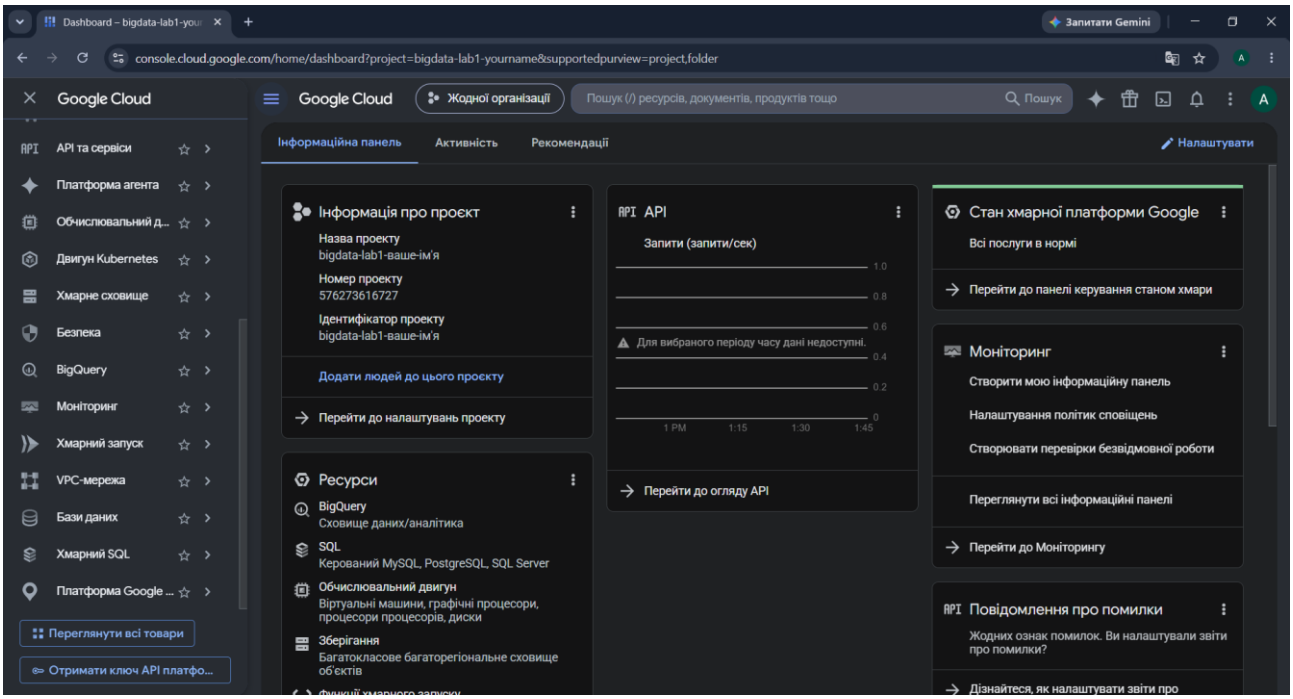


Рис. 1.3. Головна інформаційна панель (Dashboard) створеного хмарного проєкту з переліком доступних сервісів, аналітичних ресурсів та метрик стану платформи

Крок 2. Пакетне завантаження «сирих» даних (Batch Ingestion)

Ми змоделюємо класичний **Staging-рівень (Bronze Layer),** куди завантажуються необроблені та «брудні» дані у форматі Parquet.

1. Завантажте файл реальних даних поїздок Нью-Йоркського таксі за січень 2026 року (yellow_tripdata_2026-01.parquet) зі сторінки датасету на Kaggle: https://www.kaggle.com/datasets/vickyy1807/2026-nyc-yellow-taxi-gov-data

2. У лівому навігаційному меню Google Cloud Console знайдіть розділ **Analytics** та перейдіть до підрозділу **BigQuery**, відкривши середовище розробки **BigQuery Studio**, далі знайдіть ваш активний хмарний проєкт. Наведіть курсор на назву вашого проєкту, натисніть на піктограму дій у вигляді трьох вертикальних крапок (Context menu) та у випадаючому списку оберіть команду **Create dataset** (Створити набір даних).

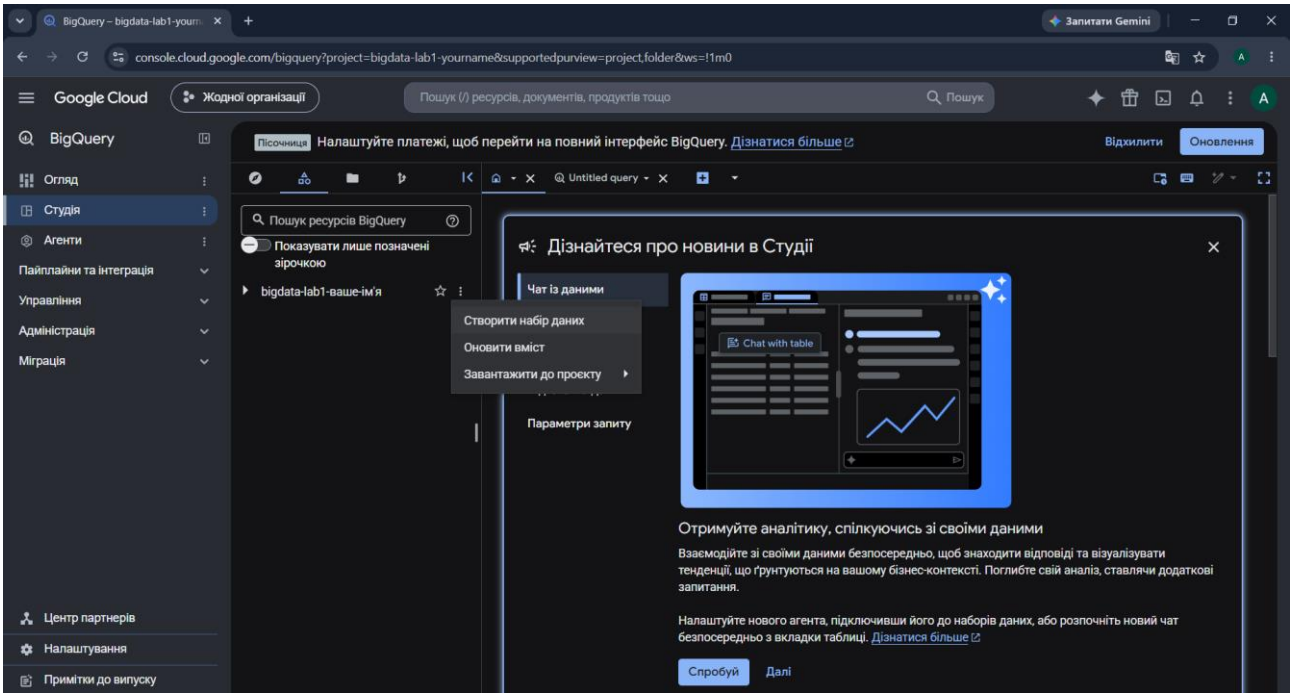


Рис. 1.4. Ініціалізація процесу створення набору даних (Dataset) через контекстне меню хмарного проєкту в інтерфейсі BigQuery Studio

Dataset ID: bronze_staging

Data location: Обов'язково оберіть US (multiple regions) або EU (multiple regions) (це важливо для доступу до публічних файлів). Натисніть **Create**.

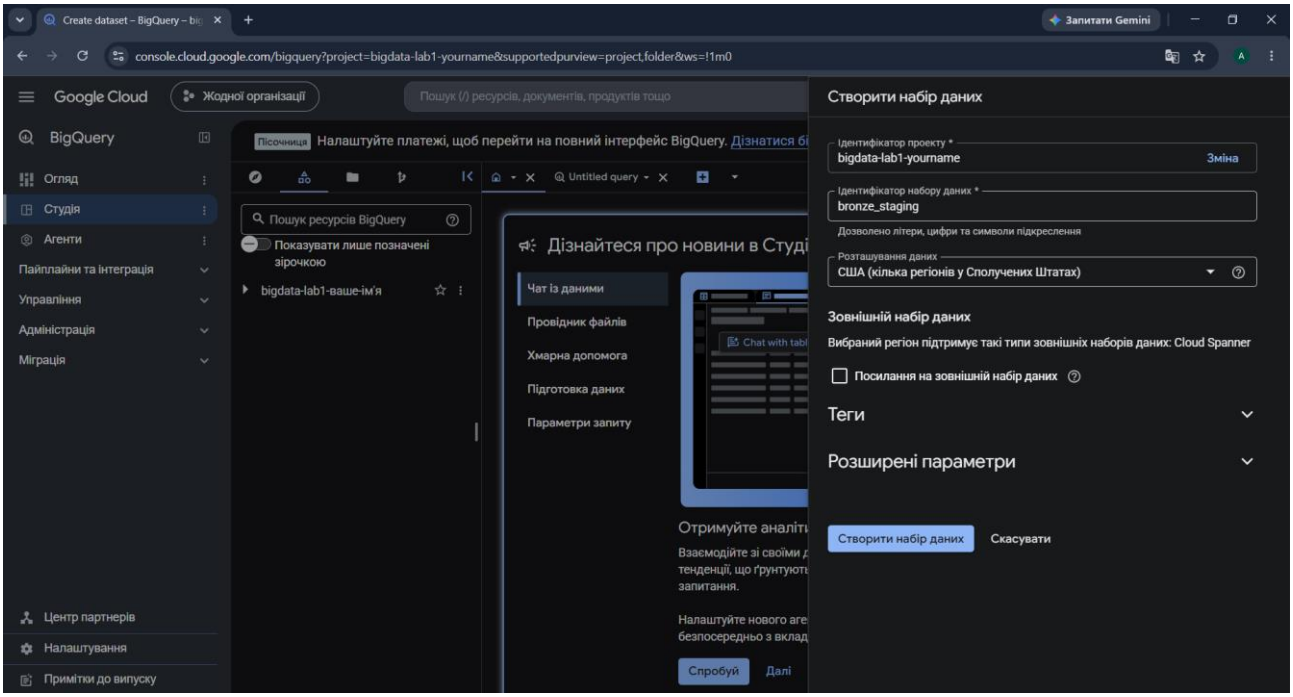


Рис. 1.5. Конфігурація ідентифікатора та вибір географічного розташування сховища (Data location) у вікні налаштувань «Створити набір даних»

3. Натисніть на три крапки біля щойно створеного датасету

bronze_staging і оберіть **Create table**.

4. У вікні створення таблиці заповніть параметри:

Create table from: Upload

Select Drive URI: Вставте посилання на ваш файл з даними у форматі parquet.

File format: PARQUET

Table name: raw_transactions

Schema: Увімкніть галочку Auto detect (BigQuery сам зчитає метадані з Parquet-файлу та сформує назви колонок, але вони будуть записані як неоптимізований «сирий» шар).

5. Натисніть **Create table**. Дані фізично скопіюються у ваш акаунт, і з ними можна працювати.

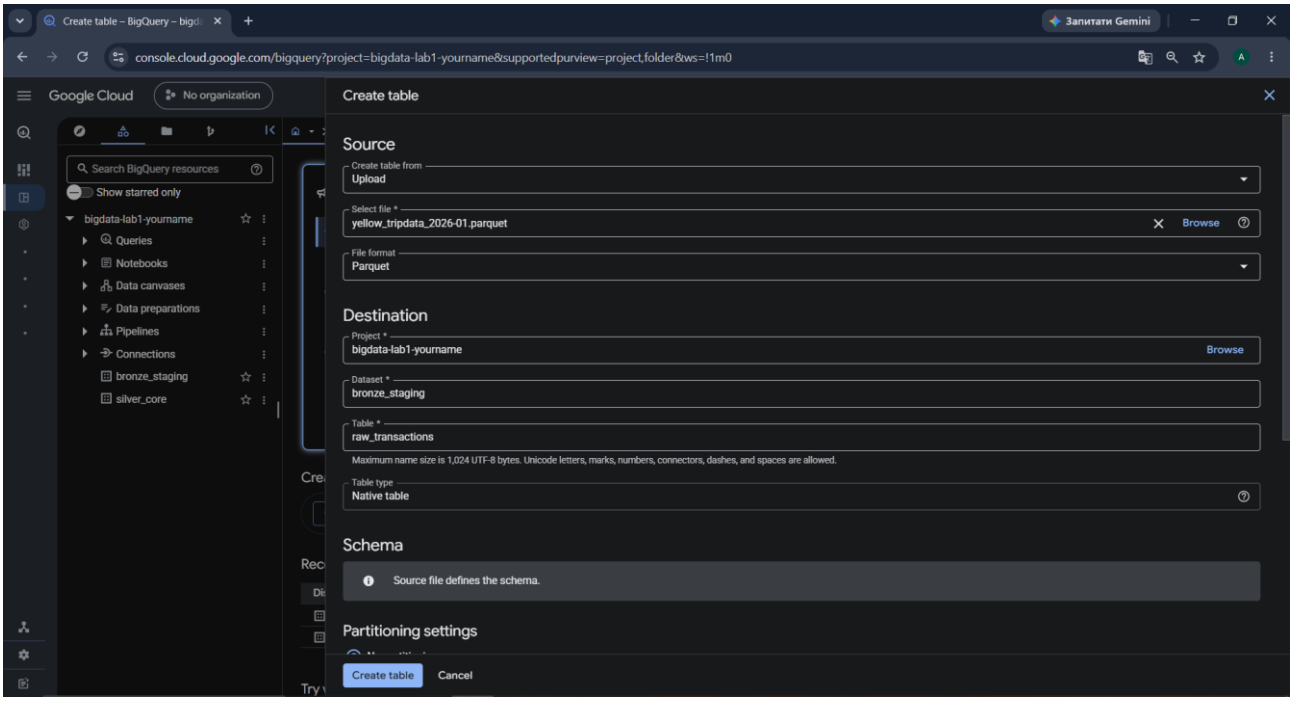


Рис. 1.6. Налаштування параметрів джерела (Source) та призначення (Destination) у вікні «Створити таблицю» під час прямого завантаження Parquet-файлу

За таким ж принципом у Google BigQuery, є можливість, створити таблицю та завантажити в неї дані можливо з таких джерел:

- **Empty table (Порожня таблиця)** — створення чистої структури таблиці вручну, куди дані будуть додаватися пізніше (через SQL-запити або API).
- **Google Cloud Storage** — пакетне завантаження файлів (CSV, Parquet, JSON, Avro), які зберігаються в хмарних сховищах-бакетах Google (gs://).
- **Upload (Завантаження)** — пряме завантаження локального

файлу (наприклад, CSV, JSON або Parquet) безпосередньо з вашого комп'ютера чи пристрою.

- **Drive (Google Диск)** — підключення файлів, які зберігаються на вашому особистому чи корпоративному Google Диску, як зовнішніх таблиць.
- **Google Bigtable** — інтеграція та читання даних у реальному часі з NoSQL бази даних Google Cloud Bigtable.
- **Amazon S3** — завантаження або підключення даних із хмарних сховищ Amazon Web Services (AWS) у межах мультихмарної аналітики BigQuery Omni.
- **Azure Blob Storage** — завантаження або підключення даних із хмарних сховищ Microsoft Azure в межах мультихмарної аналітики BigQuery Omni.
- **Existing table/view (Існуюча таблиця/представлення)** — створення копії або структури на основі вже наявних об'єктів усередині вашого проєкту BigQuery.

Після успішного формування Staging-рівня (bronze_staging) архітектура інженерії великих даних передбачає перехід до Core-рівня або шару Silver (Silver Layer). На відміну від «брудного» Bronze-шару, де дані зберігаються в первинному хаотичному вигляді, Silver-шар містить очищені, валідовані та структуровані дані, які стають «єдиним джерелом правди» (Single Source of Truth) для корпоративної аналітики.

Крок 3. Створення оптимізованої структури бази даних (DDL)

Тепер переходимо до формування **Core-рівня (Silver)**. Створимо нову таблицю, яка використовує дві головні технології Big Data: **Секціонування (Partitioning)** за датою та **Кластеризацію (Clustering)** за категоріями.

1. Створіть другий датасет із назвою silver_core (так само, як у Кроці 2, в тому ж регіоні).
2. Відкрийте чисту вкладку запитів (натисніть на значок плюсика + у верхній панелі або кнопку Create SQL query) та виконайте SQL-код, який створює порожню оптимізовану структуру:

```
CREATE OR REPLACE TABLE `silver_core.cleaned_taxi_transactions`(
  vendor_id STRING,
  pickup_datetime TIMESTAMP,
  dropoff_datetime TIMESTAMP,
  passenger_count INT64,
  trip_distance NUMERIC,
  fare_amount NUMERIC,
  tip_amount NUMERIC,
  total_amount NUMERIC,
  payment_type_id INT64
)
-- НАЛАШТУВАННЯ BIG DATA: Секціонуємо таблицю за датою подачі таксі (по днях)
PARTITION BY DATE(pickup_datetime)
-- Кластеризуємо всередині кожної секції за типом оплати та перевізником
CLUSTER BY payment_type_id, vendor_id;
```

Рис. 1.7. SQL-скрипт створення оптимізованої структури таблиці із налаштуваннями секціонування (Partitioning) та кластеризації (Clustering)

Після введення коду натисніть кнопку **Run**. У датасеті silver_core з'явиться порожня таблиця з іконкою секціонування.

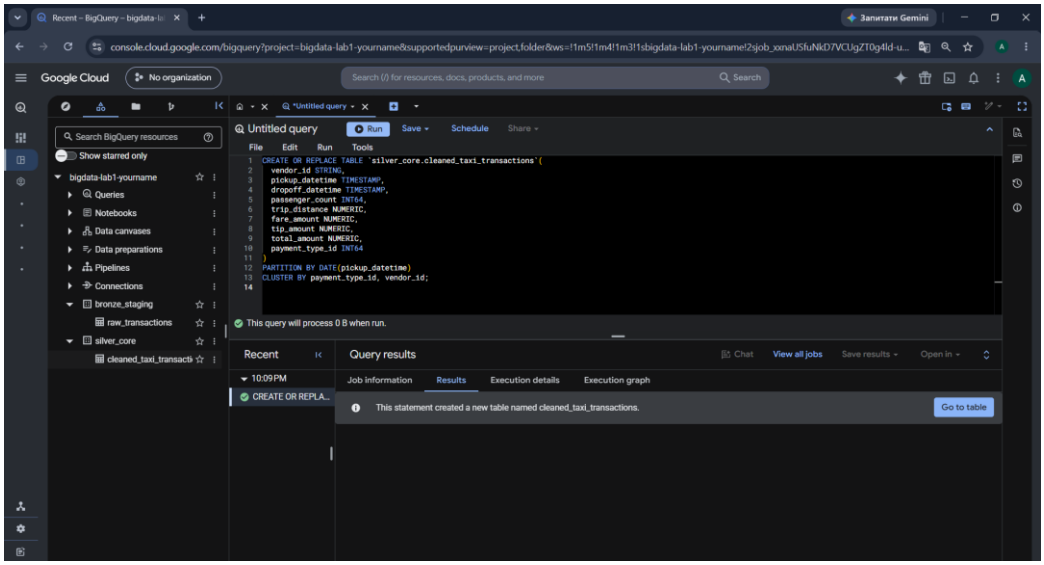


Рис. 1.8. Результат виконання DDL-скрипту створення порожньої секціонованої та кластеризованої таблиці

Крок 4. Очищення, трансформація та завантаження даних (ELT-процес за допомогою DML)

Потрібно перенести дані з «брудного» шару bronze_staging у чистий шар silver_core. Під час переносу за допомогою SQL-функцій ми виправимо помилки в типах даних, видалимо аномалії та технічні дублікати.

Відкрийте нове вікно запиту (+ Compose new query) та виконайте такий ELT-скрипт:

```
CREATE OR REPLACE TABLE `silver_core.cleaned_taxi_transactions`
PARTITION BY DATE(pickup_datetime)
CLUSTER BY payment_type_id, vendor_id AS
SELECT DISTINCT -- 1. Дедуплікація: видаляємо абсолютні дублікати рядків
   CAST(VendorID AS STRING) AS vendor_id,

   -- 2. Валідація дат: безпечно конвертуємо рядок у TIMESTAMP
   SAFE_CAST(tpep_pickup_datetime AS TIMESTAMP) AS pickup_datetime,
   SAFE_CAST(tpep_dropoff_datetime AS TIMESTAMP) AS dropoff_datetime,

   -- 3. Обробка порожніх значень: якщо кількість пасажирів NULL, ставимо дефолт 1
   COALESCE(SAFE_CAST(passenger_count AS INT64), 1) AS passenger_count,

   SAFE_CAST(trip_distance AS NUMERIC) AS trip_distance,
   SAFE_CAST(fare_amount AS NUMERIC) AS fare_amount,
   SAFE_CAST(tip_amount AS NUMERIC) AS tip_amount,
   SAFE_CAST(total_amount AS NUMERIC) AS total_amount,
   SAFE_CAST(payment_type AS INT64) AS payment_type_id
FROM `bronze_staging.raw_transactions`
-- 4. Фільтрація бізнес-аномалій та технічного сміття
WHERE tpep_pickup_datetime IS NOT NULL
  -- Обмежуємо вибірку суворо січнем 2026 року для відсікання помилкових часових міток
  AND DATE(SAFE_CAST(tpep_pickup_datetime AS TIMESTAMP)) BETWEEN '2026-01-01' AND '2026-01-31'
  AND SAFE_CAST(total_amount AS NUMERIC) > 0       -- Ціна поїздки не може бути від'ємною або нульовою
  AND SAFE_CAST(trip_distance AS NUMERIC) BETWEEN 0.1 AND 500; -- Видаляємо викиди
```

Рис. 1.9. Комплексний SQL-скрипт (DML/ELT) для очищення, трансформації, дедуплікації та перенесення масиву великих даних із шару

*Натисніть **Run**.*

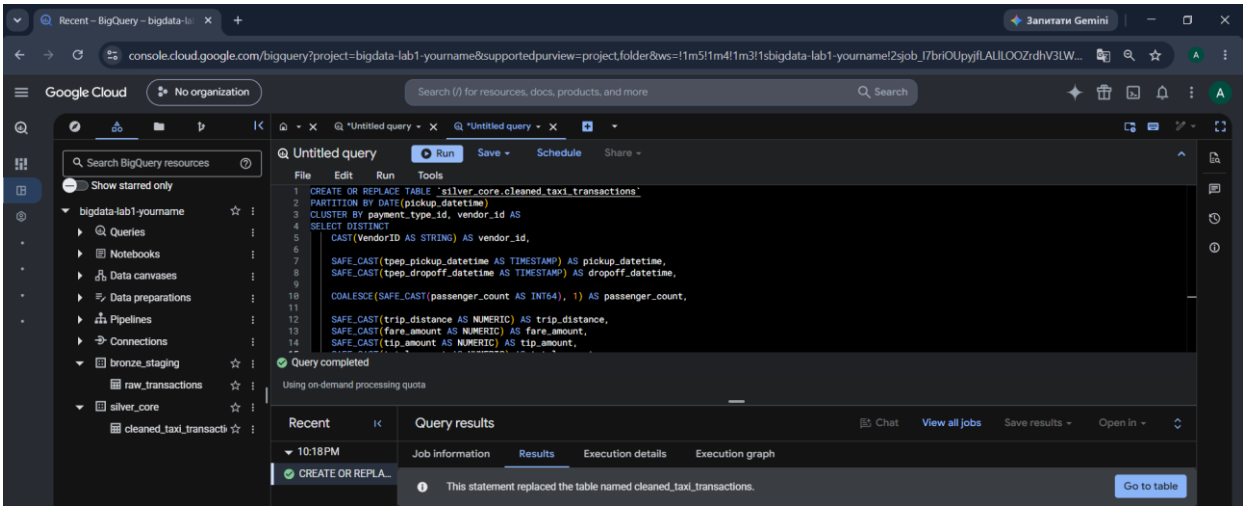


Рис. 1.10. Інтерфейс аналітичного середовища BigQuery Studio з

підтвердженням успішного завершення операції трансформації та завантаження очищених даних у таблицю

Крок 5. Складна аналітична обробка Big Data (DQL)

Коли дані очищено та структуровано, аналітики можуть проводити глибокі дослідження часових рядів. Ми використаємо **віконні функції** (OVER) мови SQL, які є стандартом для аналізу часових рядів та великих обсягів інформації.

Наприклад ми хочемо знайти для кожного перевізника (vendor_id) кумулятивний (наростаючий) підсумок зароблених чайових по днях за весь січень 2026 року.

Виконуємо наступний аналітичний запит, який наведено на рис. 1.11:

```
WITH aggregated_data AS (
  SELECT
     CAST(VendorID AS STRING) AS vendor_id,
     DATE(SAFE_CAST(tpep_pickup_datetime AS TIMESTAMP)) AS trip_date,
     SUM(SAFE_CAST(tip_amount AS NUMERIC)) AS daily_tips
  FROM `bronze_staging.raw_transactions` -- Читаємо з вашого завантаженого
файлу з Kaggle
  GROUP BY vendor_id, trip_date
)
SELECT
    vendor_id,
    trip_date,
    daily_tips,
    SUM(daily_tips) OVER (
        PARTITION BY vendor_id
        ORDER BY trip_date
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS cumulative_tips
FROM aggregated_data
WHERE trip_date BETWEEN '2026-01-01' AND '2026-01-31'
ORDER BY vendor_id, trip_date;
```

Рис. 1.11. SQL-скрипт аналітичного запиту з використанням узагальненого табличного виразу (WITH) та віконної функції (OVER) для розрахунку щоденного кумулятивного підсумку часових рядів

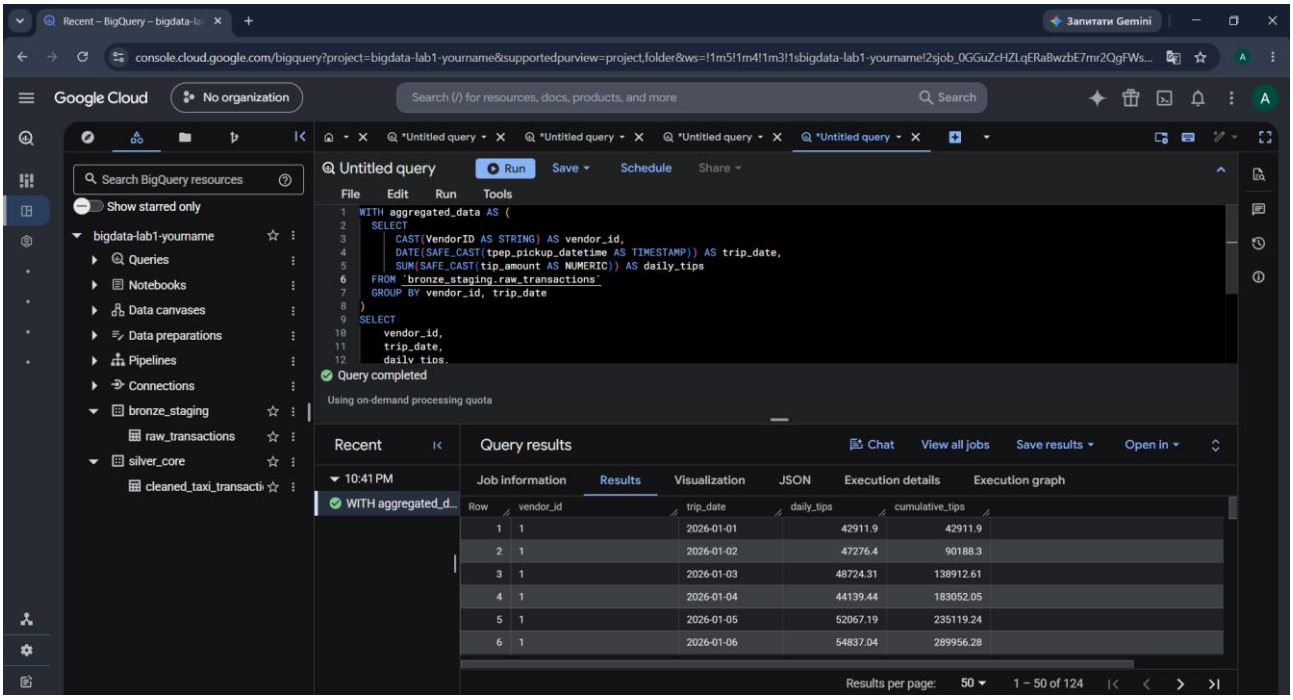


Рис. 1.12. Інтерфейс BigQuery Studio із табличним відображенням згенерованих результатів розрахунку щоденних часових рядів чайових та їхнього кумулятивного підсумку

Ознайомтеся з отриманою таблицею у вкладці Results. Ми отримали результат на 124 сторінках, по 50 записів на сторінці.

Крок 6. Аналіз продуктивності та оптимізації запитів

Оскільки за кожний запит з вашого балансу знімаються реальні кошти, однією з ключових метрик у Big Data є фінансова вартість та швидкість запиту, що залежить від кількості прочитаних гігабайтів із жорстких дисків сховища. Завдяки тому, що на Кроці 3 ми налаштували **секціонування**, ми можемо економити свої фінанси та ресурси процесора.

1. Напишіть та запустіть запит, який шукає дані за конкретний день (наприклад, за перше число місяця):

```
SELECT *
FROM `silver_core.cleaned_taxi_transactions`
WHERE pickup_datetime BETWEEN '2026-01-15 00:00:00' AND '2026-01-15 23:59:59';
```

Рис. 1.13. SQL-запит до оптимізованої таблиці Core-рівня для проведення експериментальної оцінки ефективності секціонування

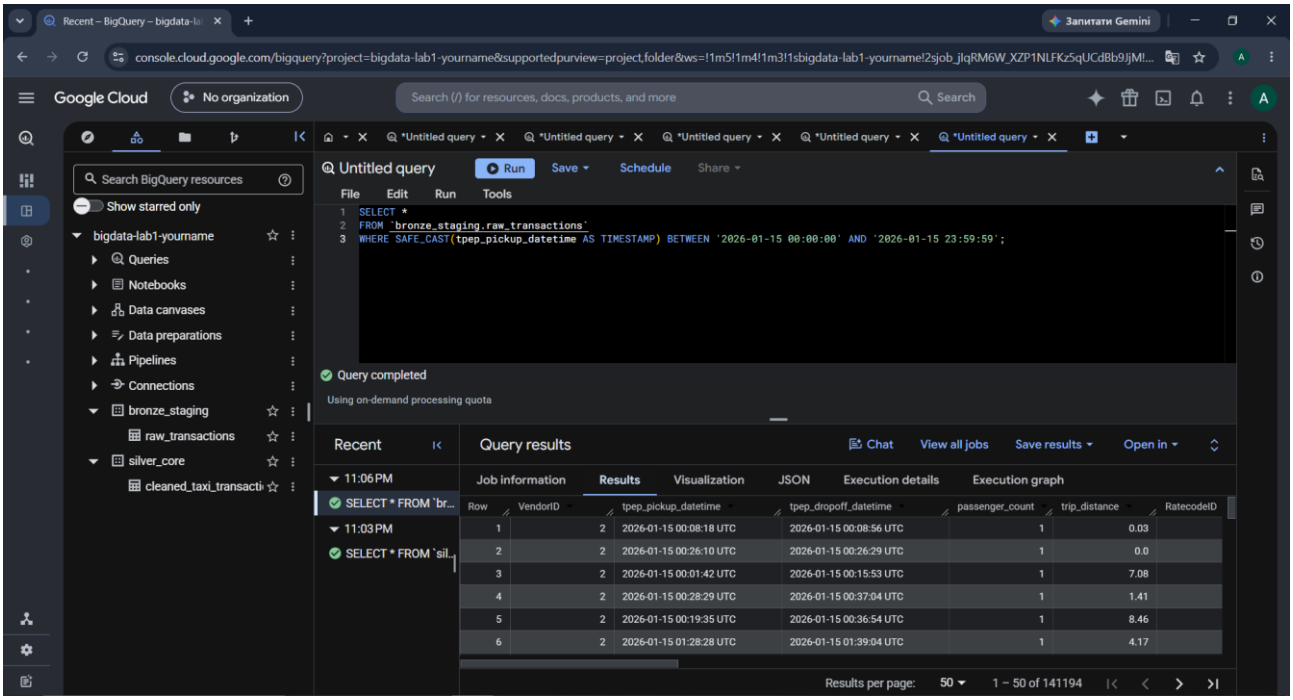


Рис. 1.14. Інтерфейс BigQuery Studio під час оцінки обсягу зчитаних даних та аудиту ефективності хмарних ресурсів

Після виконання запиту зверніть увагу на правий верхній кут редактора коду або вкладку **Job Information** / **Execution details** знизу.

Знайдіть рядок **Bytes processed** (або *This query will process...*). Запишіть у звіт, який обсяг даних прочитав цей запит.

Порівняйте це з обсягом усієї таблиці. Секціонування дозволило базі даних відсікти інші дні й сканувати лише маленький шматочок диска.

Методичні вказівки до виконання самостійної роботи:

Виконання самостійної роботи здобувачів вищої освіти (СРЗ) є прямим логічним та методологічним продовженням практичних навичок, набутих під час аудиторних лабораторних занять. Алгоритм, етапи та технологічний стек виконання самостійної роботи повністю відповідають ходу виконання завдань під час лабораторної роботи.

Для успішного вирішення індивідуальних кейсів студент повинен відтворити архітектурний підхід, опанований у лабораторному практикумі:

1. Самостійно підібрати, верифікувати та завантажити вказані масиви Big Data у **Bronze-шар** (bronze_staging) персональної хмарної пісочниці.
2. Спроєктувати оптимізовану DDL-структуру бази даних **Silver-шару** (silver_core), застосувавши відповідні стратегії секціонування (Partitioning) та кластеризації (Clustering).

3. Розробити та виконати складний **ELT-скрипт (DML)** для дедуплікації, очищення даних від технічного сміття та фільтрації бізнес-аномалій.

4. Провести аналітичну обробку та розрахувати кумулятивні чи аналітичні показники за допомогою **віконних функцій (OVER)** та узагальнених табличних виразів (WITH).

5. Здійснити фінансово-ресурсний аудит запитів (**Bytes processed**) та оформити результати у вигляді презентації або аналітичного звіту (есе).

ВАРІАНТИ КЕЙСОВИХ ЗАВДАННЬ ДЛЯ САМОСТІЙНОГО ВИРІШЕННЯ:

Кейсове завдання 1. Аналіз ризиків та ліквідності у сфері мікрокредитування (FinTech)

МФО (мікрофінансова організація) масово видає короткострокові позики. Через високу динаміку мінливого середовища компанія стикається зі зростанням частки неповернених кредитів (NPL). Необхідно виявити приховані тренди в поведінці боржників.

Вхідні дані (Bronze): Датасет транзакцій та прострочень по мікрокредитах (обсяг > 2 млн записів).

Специфікація Silver-шару:

секціонування: за датою видачі кредиту (loan_date);

кластеризація: за кредитним рейтингом клієнта (credit_score_band) та статусом прострочення (default_status).

Необхідно очистити дані від помилкових системних дублікатів. Написати SQL-запит із віконною функцією для розрахунку кумулятивної суми втрат від неповернених кредитів окремо для кожної категорії кредитного рейтингу протягом часового ряду.

Кейсове завдання 2. Оптимізація ланцюгів постачання та логістики в E-commerce.

Великий міжнародний маркетплейс аналізує швидкість доставки товарів. В умовах логістичних криз важливо виявити «вузькі місця» (bottlenecks) — регіони та перевізників, які найчастіше порушують терміни доставки.

Вхідні дані (Bronze): Масив даних про трекінг відправлень, що містить часові мітки замовлення, відвантаження та фінальної доставки (обсяг > 1.5 млн записів).

Специфікація Silver-шару:

секціонування: за датою створення замовлення (order_date);

кластеризація: за кодом країни призначення (country_code) та ідентифікатором логістичного провайдера (carrier_id.

Необхідно обробити пропущені значення NULL у датах доставки (замовлення, які ще в дорозі або втрачені). За допомогою віконних функцій розрахувати ковзне середнє значення (Moving Average) тривалості доставки товарів у розрізі кожного перевізника для відстеження динаміки погіршення чи покращення сервісу.

Кейсове завдання 3. Виявлення фроду (шахрайства) у банківських транзакціях.

Роздрібний банк прагне мінімізувати збитки від несанкціонованих операцій з платіжними картками. Необхідно побудувати аналітичний скрипт, який маркує підозрілі транзакції на основі лімітів та частоти операцій.

Вхідні дані (Bronze): Потік транзакцій клієнтів банку (обсяг > 5 млн записів).

Специфікація Silver-шару:

секціонування: за датою здійснення транзакції (transaction_date);

кластеризація: за типом операції (terminal_type) та категорією торгової точки (mcc_code).

Необхідно провести жорстку фільтрацію аномалій (транзакції з від'ємними сумами, некоректні коди валют). Використовуючи аналітичні віконні функції, розрахувати для кожного клієнта сумарний обсяг витрат за останні 3 транзакції поспіль та порівняти його із середньодобовим показником для виявлення різких стрибків (ознака компрометації картки).

Кейсове завдання 4. Аналіз поведінки споживачів (Customer Analytics) в ритейлі.

Мережа супермаркетів запустила нову програму лояльності. Менеджменту потрібно оцінити її ефективність, проаналізувавши щоденні витрати постійних клієнтів та динаміку зростання середнього чека.

Вхідні дані (Bronze): Історія чеків та куплених товарів програми лояльності (обсяг > 3 млн записів).

Специфікація Silver-шару:

секціонування: за датою друку чека (sales_date);

кластеризація: за ідентифікатором магазину (store_id) та категорією товару (product_category).

Необхідно очистити датасет від технічних повернень товару (чеки з

нульовою підсумковою сумою). Написати SQL-запит з використанням CTE та віконних функцій для визначення кумулятивного відсотка внеску кожної категорії товарів у загальний щоденний прибуток конкретного торгового центру.

Кейсове завдання 5. Моніторинг енергоспоживання та тарифна оптимізація (Green Economy)

Енергетична компанія аналізує дані з розумних лічильників (Smart Meters) промислових підприємств для оптимізації навантаження на енергомережу в пікові години та розрахунку гнучких екологічних тарифів.

Вхідні дані (Bronze): Погодинні логи споживання електроенергії об'єктами (обсяг > 10 млн записів).

Специфікація Silver-шару:

секціонування: за датою зчитування показників (reading_date);

кластеризація: за типом промислового сектора (industry_type) та ідентифікатором зони енергомережі (grid_zone_id).

Необхідно реалізувати ELT-трансформацію із безпечною конвертацією типів спожитої потужності за допомогою SAFE_CAST. Використовуючи віконні функції, ранжувати (DENSE_RANK) години максимального споживання для кожного підприємства з метою виявлення періодів найбільшого навантаження на інфраструктуру.

Тема 2 «Сучасні програмні застосунки обробки Big Data для вирішення професійних задач в сфері економіки»

Лабораторна робота "Практичне знайомство з сучасними ППП для аналізу Big Data"

Мета роботи - набути практичних навичок статистичного аналізу вибіркових даних і побудови множинної лінійної регресійної моделі засобами надбудови «Аналіз даних» у MS Excel.

Завдання лабораторної роботи:

сформувати вихідний масив, розрахувати параметри описової статистики для кожної змінної та побудувати двофакторне рівняння регресії;

записати рівняння моделі навчального прикладу та перевірити його показники на відповідність еталонним контрольним значенням;

сформувати графіки залишків та графік нормальної ймовірності для

оцінки виконання передумов методу найменших квадратів (МНК).

Методичні рекомендації до виконання лабораторної роботи

1. Підключення надбудови «Аналіз даних»

Файл (File) → Параметри (Options) → Надбудови (Add-ins) → Керування: Надбудови Excel → Перейти → Analysis ToolPak

Після підключення на вкладці Дані (Data) має бути доступна команда Аналіз даних (Data Analysis).

2. Вихідні дані

Необхідно дослідити статистичний зв'язок між прибутком підприємств (Y), випуском продукції (X1) та обсягом роздрібного товарообороту (X2). Вихідний масив із 20 спостережень збережено без змін. Вхідна інформація наведена в бал. 2.1.

Таблиця 2.1

Вхідна інформація

№ підприємства	Прибуток підприємства, млн. грн (У)	Випуск продукції, тис. грн (X1)	Обсяг роздрібного товарообороту підприємств, млн.грн. (X2)
1	6	40	30
2	4,6	20	33
3	4,4	31	20
4	4,5	32	25
5	5,5	34	29
6	4,8	35	20
7	5,1	37	21
8	5,2	32	20
9	7	39	35
10	5,3	35	30
11	7,5	50	35
12	7,7	37	30
13	7,3	50	40
14	7	38	42
15	6,7	50	39
16	5,7	35	35
17	6	46	36
18	6,4	49	38
19	7,1	54	41
20	6,3	45	34

3. Описова статистика

Дані (Data) → Аналіз даних (Data Analysis) → Описова статистика (Descriptive Statistics)

1. У полі Input Range задайте числові стовпці Y, X1, X2 разом із заголовками.

2. Оберіть Grouped by: Columns та Labels in first row.

3. Оберіть Output Range або New Worksheet Ply.

4. Установіть Summary statistics та натисніть OK.

5. Для кожної змінної проаналізуйте середнє, медіану, стандартне відхилення, мінімум, максимум, асиметрію та ексцес.

Для контролю правильності обчислень у цьому наборі: середнє Y = 6,005; X1 = 39,45; X2 = 31,65.

Стандартні відхилення становлять приблизно 1,058; 8,507; 7,250 відповідно. Очікуваним результатом буде таблиця описової статистики та 3–5 речень, що характеризують центр, варіацію та діапазон кожної змінної.

4. Побудова множинної лінійної регресії

Дані (Data) → Аналіз даних (Data Analysis) → Регресія (Regression)

1. Input Y Range: стовпець «Прибуток підприємства, млн грн (Y)».

2. Input X Range: два суміжні стовпці X1 та X2.

3. Якщо виділено заголовки, установіть Labels.

4. Установіть Confidence Level 95%, якщо потрібні 95% довірчі інтервали.

5. Для діагностики позначте Residuals, Standardized Residuals, Residual Plots та Normal Probability Plots.

6. Оберіть місце для результату та натисніть OK.

5. Інтерпретація результатів

Для наведеного масиву Excel необхідно дати результати, узгоджені з контрольними значеннями, наведеними в табл. 2.2.

Таблиця 2.2

Результати регресійного аналізу

Показник	**Контрольне значення**
Multiple R	≈ 0,814
R Square	≈ 0,662
Adjusted R Square	≈ 0,622
F	≈ 16,65
Significance F	≈ 0,000099
Intercept	≈ 1,691
X1	≈ 0,0563
X2	≈ 0,0662

Таким чином, рівняння множинної регресії має наступний вигляд:

Y = 1,691 + 0,0563·X1 + 0,0662·X2

R^2 ≈ 0,662 означає, що близько 66,2 % варіації Y у цій вибірці описується моделлю з X1 та X2. Це не означає, що фактори «спричиняють» 66,2 % зміни прибутку.

Загальний F-тест перевіряє нульову гіпотезу про спільну статистичну незначущість коефіцієнтів при предикторах. За Significance F≈ 0,000099 модель статистично значуща в цілому на рівні 0,05.

Для окремих коефіцієнтів у цьому прикладі p-value становлять приблизно 0,0214 для X1 та 0,0212 для X2, тому обидва предиктори статистично значущі на рівні 0,05. Константа також має p ≈ 0,0406.

Коефіцієнт при X1 інтерпретується так: за незмінного X2 збільшення X1 на одну одиницю пов’язане зі збільшенням прогнозованого Y приблизно на 0,0563 одиниці. Аналогічно для X2 — приблизно на 0,0662 одиниці за незмінного X1.

6. Аналіз залишків

Для лінійної регресії недостатньо лише отримати R^2 та p-value. Необхідно також проаналізувати залишки та графіки, сформовані Excel. Перевірити графік Residuals vs X1 та Residuals vs X2: залишки не повинні утворювати очевидний систематичний рисунок.

Семінарське заняття "Порівняльна характеристика сучасних програмних застосунків для аналізу Big Data: SPSS, EViews, R, Tableau, Python, Tableau, PowerBI"

Написання есе у вигляді презентації є самостійною творчою та аналітичною роботою здобувача вищої освіти. Метою семінарського заняття є теоретичне осмислення сутності сучасного інструментарію Data Science, критичний аналіз закономірностей функціонування аналітичних платформ, а також розвиток навичок структурування складної технічної інформації, візуального моделювання архітектури великих даних та публічного захисту власних аналітичних тез перед аудиторією.

Особливістю виконання семінарського заняття є порівняльний аналіз архітектурних можливостей сучасного софту для обробки надвеликих масивів інформації. Здобувач має продемонструвати, як використання конкретних методів (пакетної обробки, віконних функцій, економетричного моделювання та інтерактивного дашбордингу) дозволяє трансформувати мільйонні потоки «сирих» даних у структуровані

аналітичні інсайти для бізнесу, автоматизувати виявлення прихованих трендів та мінімізувати ризики помилок під час стратегічного планування в мінливому економічному середовищі.

Кожен здобувач вищої освіти самостійно обирає один з сучасних програмних застосунків для аналізу Big Data, детально аналізує його, визначає переваги та недоліки та трансформує результати свого дослідження у аналітичне есе-презентацію.

Вимоги до есе у вигляді презентації. Презентація повинна замінити класичний текст есе, зберігаючи його логіку та складатися обсягом не менше 8 сторінок. Формат презентації у вигляді PPTX (MS PowerPoint) або PDF. Здобувач публічно на семінарському занятті презентує своє есе.

Методичні вказівки до виконання самостійної роботи:

Вирішення прикладних кейсових завдань, висунутих на самостійне опрацювання, повністю відповідає алгоритму виконання, технологічному стеку, послідовності команд хмарного/локального табличного процесора MS Excel, а також логіці інтерпретації результатів, які були детально опановані та продемонстровані під час вирішення завдання на лабораторній роботі.

ВАРІАНТИ КЕЙСОВИХ ЗАВДАННЬ ДЛЯ САМОСТІЙНОГО ВИРІШЕННЯ:

Кейсове завдання 1. Дослідження факторів капіталізації та ринкової вартості технологічних компаній

Інвестиційний аналітик досліджує ринок акцій в умовах високої волатильності. Необхідно оцінити, як фінансові драйвери впливають на ринкову капіталізацію високотехнологічних підприємств.

Масив даних формує здобувач на основі офіційних статистичних джерел. Вибірка повинна мати осяг не менше 30 компаній з наступними факторами:

залежна змінна (Y): Ринкова капіталізація компанії (млн дол.);

фактор 1 (X_1): Чистий річний прибуток (млн дол.);

фактор 2 (X_2): Річні витрати на науково-дослідні та дослідно-конструкторські роботи — R&D (млн дол.);

фактор 3 (X_3): Загальна кількість активних користувачів платформи / клієнтів (млн осіб).

Необхідно побудувати багатофакторну регресію. Визначити, яка саме складова — фінансовий результат чи інвестиції в інновації (R&D) — є статистично значущим драйвером вартості бізнесу в сучасній цифровій економіці.

Кейсове завдання 2. Аналіз ефективності витрат на маркетинг та рекламу в E-commerce.

Директор з маркетингу міжнародного маркетплейса прагне оптимізувати рекламний бюджет в умовах жорсткої конкуренції. Потрібно встановити математичну залежність між каналами просування та фінальними продажами.

Масив даних формує здобувач на основі офіційних статистичних джерел. Вибірка повинна мати осяг не менше 24 регіональні філії з наступними факторами:

залежна змінна (Y): щомісячний обсяг продажів магазину (тис. грн);

фактор 1 (X_1): витрати на контекстну та пошукову рекламу в Google (тис. грн);

фактор 2 (X_2): бюджет на залучення блогерів та таргетинг у соціальних мережах (тис. грн);

фактор 3 (X_3): обсяг витрат на email-маркетинг та програми лояльності (тис. грн).

Необхідно розрахувати дескриптивні метрики коливання продажів. Побудувати модель множинної регресії. Оцінити коефіцієнт детермінації (R^2) та визначити канал просування, що забезпечує найбільший маржинальний приріст продажів на кожну вкладену гривню.

Кейсове завдання 3. Оцінка інвестиційної привабливості та вартості комерційної нерухомості

Девелоперська компанія проводить аудит об'єктів комерційної нерухомості в мегаполісі для формування гнучкої цінової політики. Необхідно змоделювати ринкову вартість офісних центрів на основі їхніх просторових та технічних характеристик.

Масив даних формує здобувач на основі офіційних статистичних джерел. Вибірка повинна мати осяг не менше 35 об’єктів нерухомості з наступними факторами:

залежна змінна (Y): ринкова оціночна вартість офісного центру (млн грн);

фактор 1 (X_1): загальна корисна площа об'єкта (кв. м).;

фактор 2 (X_2): відстань від центрального ділового району міста (км);

фактор 3 (X_3): кількість паркувальних місць на території комплексу (од.).

Необхідно побудувати лінійну модель. Перевірити константу та коефіцієнти на статистичну значущість. Дослідити графіки залишків, щоб переконатися, що віддаленість від центру міста не викликає систематичного викривлення похибок моделі (тест на гетероскедастичність).

Кейсове завдання 4. Моделювання обсягів імпорту продукції в умовах макроекономічної мінливості.

Аналітичний департамент Міністерства економіки досліджує стійкість платіжного балансу країни. Метою є оцінка того, як внутрішні макроекономічні чинники моделюють попит на імпортні товари та послуги.

Масив даних формує здобувач на основі офіційних статистичних джерел. Вибірка повинна мати осяг не менше 28 звітних кварталів з наступними факторами:

залежна змінна (Y): загальний обсяг імпорту в країну (млн грн);

фактор 1 (X_1): реальний валовий внутрішній продукт — ВВП (млн грн);

фактор 2 (X_2): офіційний середньоквартальний курс національної валюти щодо долара США;

фактор 3 (X_3): реальні наявні грошові доходи населення (млн грн).

Необхідно провести розвідувальний аналіз часових рядів. Оцінити модель множинної регресії. Проаналізувати Significance F. Сформулювати застереження щодо наявності мультиколінеарності (взаємозв'язку між ВВП та доходами населення) та її впливу на зміщення оцінок коефіцієнтів.

Кейсове завдання 5. Дослідження факторів продуктивності праці та кадрового потенціалу підприємства

HR-директор великої виробничої корпорації проводить реінжиніринг бізнес-процесів. Для підвищення ефективності персоналу необхідно встановити, які саме інвестиції в людський капітал забезпечують найвищу віддачу.

Масив даних формує здобувач на основі офіційних статистичних джерел. Вибірка повинна мати осяг не менше 32 цехи/ виробничі дільниці з наступними факторами:

залежна змінна (Y): середня продуктивність праці (обсяг виробленої продукції на одного робітника, тис. грн/особу);

фактор 1 (X_1): середній рівень заробітної плати на дільниці (грн);

фактор 2 (X_2): інвестиції корпорації в автоматизацію та оновлення обладнання дільниці (тис. грн);

фактор 3 (X_3): середній стаж роботи / досвід персоналу на даному виробництві (років).

Необхідно розрахувати середні, мінімальні та максимальні рівні продуктивності праці через ППП «Описова статистика». Розробити багатофакторне економетричне рівняння. За допомогою значення Adjusted R Square обґрунтувати, яка частка коливань продуктивності праці пояснюється обраними факторами мотивації та модернізації.

Змістовий модуль 2. Методи і моделі інтелектуального аналізу Big Data в економіці

Тема 3. Методи нечіткої логіки в аналізі Big Data в управлінні бізнес-проєктами та системами

Лабораторна робота "Використання генетичних алгоритмів в Big Data"

Мета роботи – набуття глибоких практичних навичок проектування інтелектуальних систем підтримки прийняття рішень та крос-платформеної інженерії великих даних. Здобувачі мають навчитися інтегрувати хмарні сховища даних (Google BigQuery) з аналітичними середовищами розробки (Google Colab) за допомогою інструментів програмного інтерфейсу (API).

Завдання лабораторної роботи:

налаштувати робочий простір у хмарному сховищі Google Диск та ініціалізувати інтерактивний блокнот Google Colaboratory для написання та виконання коду Python;

еалізувати процедуру віддаленої хмарної автентифікації користувача (auth.authenticate_user()) для встановлення безпечного з'єднання з GCP;

розгорнути аналітичний стек бібліотек Python, встановивши спеціалізовані пакети для еволюційних обчислень (deap), нечіткої логіки (scikit-fuzzy), обробки Big Data (google-cloud-bigquery, pandas) та візуалізації (matplotlib).

Методичні рекомендації до виконання лабораторної роботи

Кейсове завдання. Сучасні великі транспортні та логістичні компанії (на кшталт Uber, Bolt, Uklon) щодня генерують мільйони записів про поїздки. Головними викликами для управління такими бізнес-проєктами є висока невизначеність ринкових факторів (резькі коливання попиту, затори, погодні умови) та обмеженість ресурсів (лімітована кількість автомобілів у парку). Звичайний статичний підхід до формування тарифів та виводу водіїв на лінії призводить або до простою машин і збитків (перенасичення ринку), або до втрати клієнтів та недоотримання прибутку (дефіцит машин у часи пік).

Необхідно побудувати інтелектуальну систему динамічного ціноутворення та еволюційної оптимізації ресурсів цифрового таксопарку на основі технологій Big Data.

Крок 1. Перехід на Google Диск та створення папки

Найзручніше створювати файли Colab безпосередньо через інтерфейс Google Диску, щоб ви завжди знали, в якій папці лежить ваша лабораторна робота.

1. Відкрийте браузер та перейдіть на google.com.
2. Увійдіть у свій Google-акаунт.
3. Натисніть кнопку **+ Створити** (New) у верхньому лівому куті.
4. Оберіть **Папка** (New folder), назвіть її, наприклад, Лабораторні Big Data, та натисніть **Створити**.
5. Зайдіть у щойно створену папку подвійним кліком.

Крок 2. Створення файлу Google Colab всередині папки

1. Знаходячись у вашій папці на Диску, знову натисніть кнопку **+ Створити** у лівому куті.
2. Наведіть курсор на пункт **Більше** (More) внизу списку.

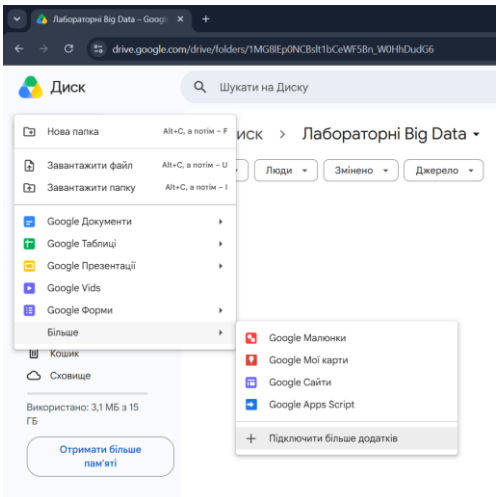


Рис. 3.1. Виклик додаткового меню Google Workspace Marketplace для підключення середовища розробки Google Colaboratory до хмарного сховища

3. У додатковому меню знайдіть і натисніть там же "+Підключити більше додатків", введіть у пошуку "Colaboratory" та встановіть.

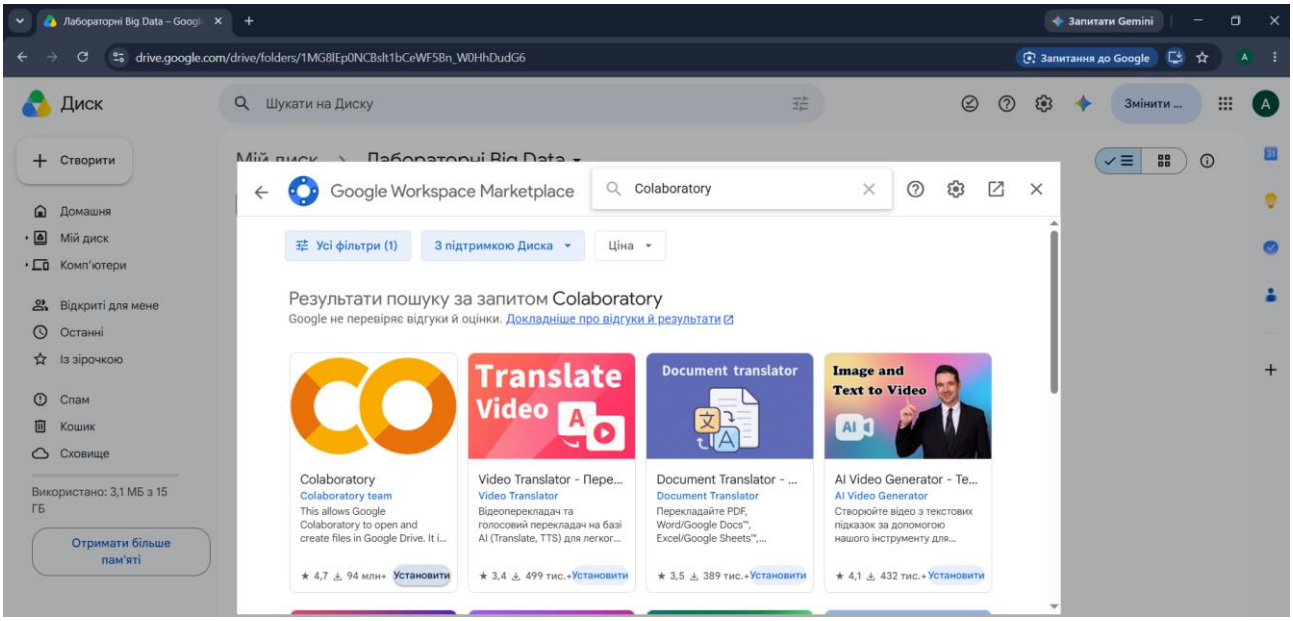


Рис. 3.2. Пошук та ініціалізація процесу встановлення хмарного інтерактивного блокнота Google Colaboratory через інтерфейс Google Workspace Marketplace

4. У додатковому меню знайдіть і натисніть **Google Colaboratory.**

5. У новій вкладці браузера автоматично відкриється ваш новий чистий блокнот (файл) Python.

6. У верхньому лівому куті натисніть на назву за замовчуванням (Untitled0.ipynb).

7. Перейменуйте файл на наприклад, **BigData_Lab2_YourName.ipynb**.

8. Зверніть увагу: тепер будь-які зміни, коди чи графіки, які ви будете писати, **автоматично зберігаються** у цей файл на вашому Google Диску (як у звичайному Google Документі).

Крок 3. Підключення бібліотек

Тепер у першу порожню клітинку Colab вставте код для встановлення всіх необхідних інструментів для нашої лабораторної роботи (генетичні алгоритми deap та нечітка логіка skfuzzy):

```
!pip install deap scikit-fuzzy google-cloud-bigquery pandas matplotlib
```

Натисніть кнопку **Виконати клітинку** (Run) ліворуч від клітинки або натисніть комбінацію клавіш **Shift + Enter**. Система почне встановлення.

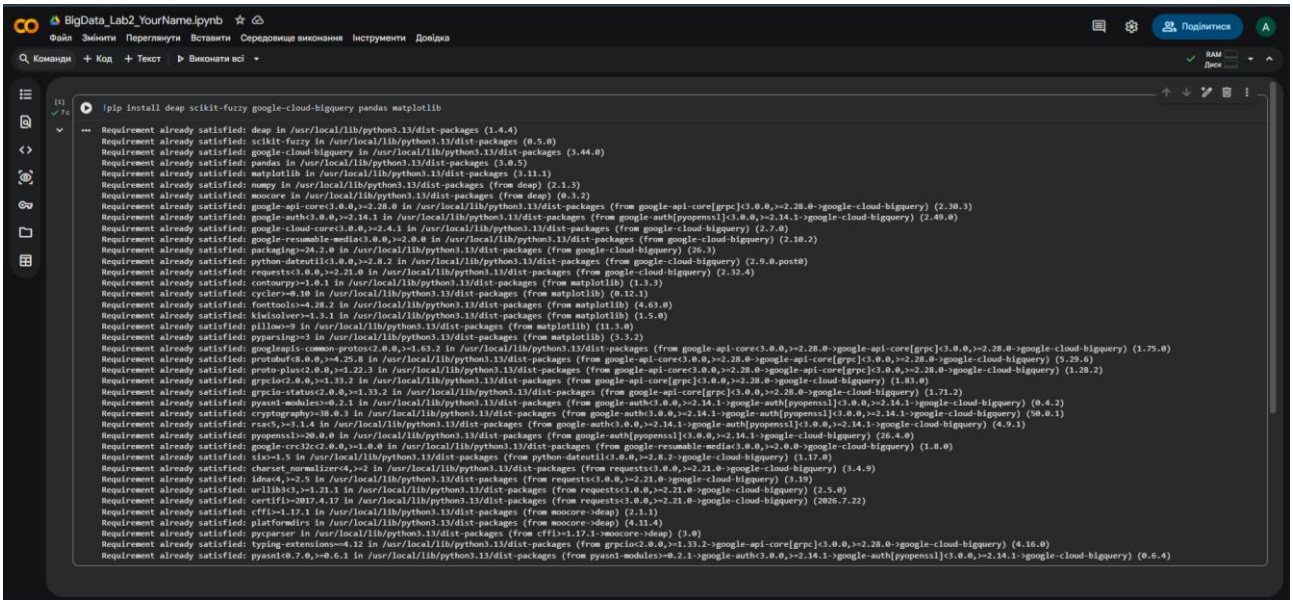


Рис. 3.3. Процес інсталяції та перевірки залежностей бібліотек штучного інтелекту, еволюційного моделювання та хмарної аналітики в інтерактивній клітинці Google Colab

Крок 4. Інтеграція BigQuery та Python (Зв'язок з Лабораторною №1)

Замість генерації випадкових чисел, ми дістанемо реальні агреговані метрики за січень 2026 року, які ми очистили в минулій роботі. Нам потрібні: середня дистанція поїздок, середній тариф та інтенсивність попиту по днях.

Знайдіть та натисніть угорі зліва **+Код.** Створюємо нову клітинку і вносимо в неї наступний код, що допоможе нам підключитися до вашого проєкту BigQuery та вивантажити дані:

```
from google.colab import auth
auth.authenticate_user()
print("Авторизація пройшла успішно!")
```

Після запуску з'явиться вікно безпеки. Якщо треба увійди під тим самим Google-акаунтом, де ти створював проєкт у Лабораторній №1 та вибравши усі необхідні «галочки», кликни "Дозволити".

У разі коли ви все зробили вірно, то ми отримаємо за результатом виконання клітинки напис **Авторизація пройшла успішно!**

```
from google.cloud import bigquery
import pandas as pd

# УВАГА: Заміни 'bigdata-lab1-yourname' на точний ID свого проєкту з Лабораторної №1
PROJECT_ID = 'bigdata-lab1-yourname'

client = bigquery.Client(project=PROJECT_ID)

# SQL-запит до сирої таблиці bronze з першої лабороторної работи
query = """
SELECT
    -- Якщо дата некоректна, групуємо просто по днях наявних даних
    COALESCE(DATE(SAFE_CAST(tpep_pickup_datetime AS TIMESTAMP)), CURRENT_DATE()) AS trip_date,
    COUNT(*) AS total_trips,
    AVG(SAFE_CAST(trip_distance AS NUMERIC)) AS avg_distance,
    AVG(SAFE_CAST(fare_amount AS NUMERIC)) AS avg_fare
FROM `bronze_staging.raw_transactions`
GROUP BY trip_date
ORDER BY trip_date
LIMIT 31; -- обмежуємося 31 днем для роботи алгоритму
"""

# Завантажуємо результат напряму в датафрейм Python
df = client.query(query).to_dataframe()

print("Перевірка завантаження даних:")
print(f"Отримано рядків: {len(df)}")
print(df.head()) # Покаже перші 5 записів
```

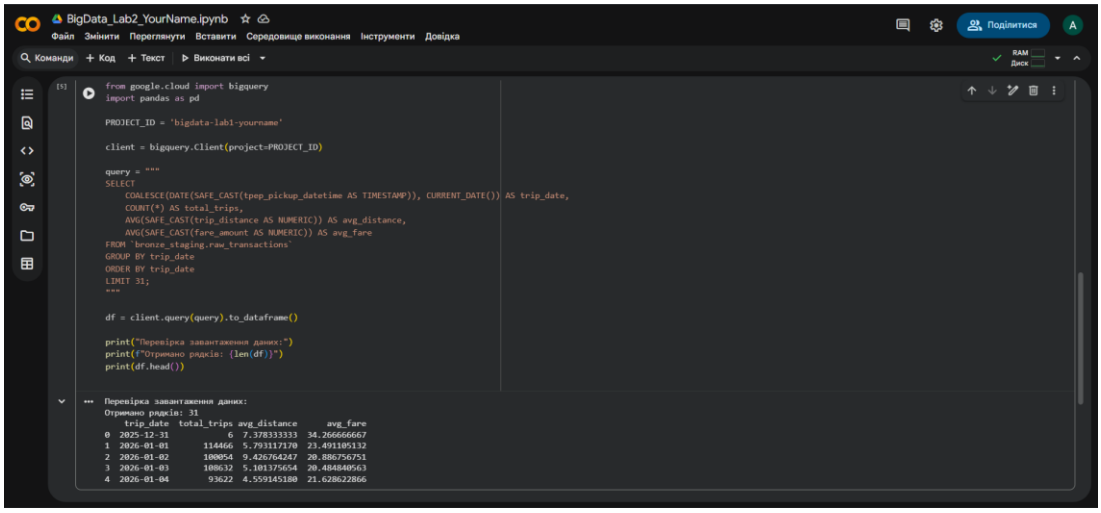


Рис. 3.4. Реалізація API-запиту мовою Python для імпорту та агрегування великого масиву транзакційних даних із хмарного сховища Google BigQuery в структуру Pandas DataFrame

Крок 5. Моделювання бізнес-проєкту: Система нечіткої логіки (Fuzzy Logic)

Для управління бізнес-проєктом таксопарку нам потрібно визначати **Оптимальну націнку (Surge Multiplier)** на основі двох сирих чинників з Big Data:

1. **Попит (Demand)** (кількість поїздок за день).
2. **Завантаженість (Traffic/Distance)** (середня дистанція/тривалість).

Пропишемо логіку нечітких правил (якщо попит *ВИСОКИЙ* і дистанція *КОРОТКА*, то націнка *ВИСОКА*, щоб залучити водіїв у центр міста).

```
import numpy as np
import skfuzzy as fuzzy
from skfuzzy import control as ctrl

# Описуємо нечіткі змінні: Входи (Попит, Дистанція) та Вихід (Націнка)
demand = ctrl.Antecedent(np.arange(0, 200000, 5000), 'demand')
distance = ctrl.Antecedent(np.arange(0, 15, 0.5), 'distance')
surge = ctrl.Consequent(np.arange(1.0, 4.0, 0.1), 'surge')

# Автоматичне або ручне налаштування трикутних функцій приналежності (trimf)
demand['low'] = fuzzy.trimf(demand.universe, [0, 0, 80000])
demand['medium'] = fuzzy.trimf(demand.universe, [40000, 100000, 160000])
demand['high'] = fuzzy.trimf(demand.universe, [120000, 200000, 200000])

distance['short'] = fuzzy.trimf(distance.universe, [0, 0, 4])
distance['medium'] = fuzzy.trimf(distance.universe, [2, 6, 10])
distance['long'] = fuzzy.trimf(distance.universe, [7, 15, 15])

surge['normal'] = fuzzy.trimf(surge.universe, [1.0, 1.0, 1.8])
surge['medium'] = fuzzy.trimf(surge.universe, [1.5, 2.2, 3.0])
surge['high'] = fuzzy.trimf(surge.universe, [2.5, 4.0, 4.0])

# База експертних правил для бізнес-системи
rule1 = ctrl.Rule(demand['high'] & distance['short'], surge['high'])
rule2 = ctrl.Rule(demand['medium'], surge['medium'])
rule3 = ctrl.Rule(demand['low'], surge['normal'])

surge_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
surge_sim = ctrl.ControlSystemSimulation(surge_ctrl)
print("Систему нечіткої логіки налаштовано.")
```

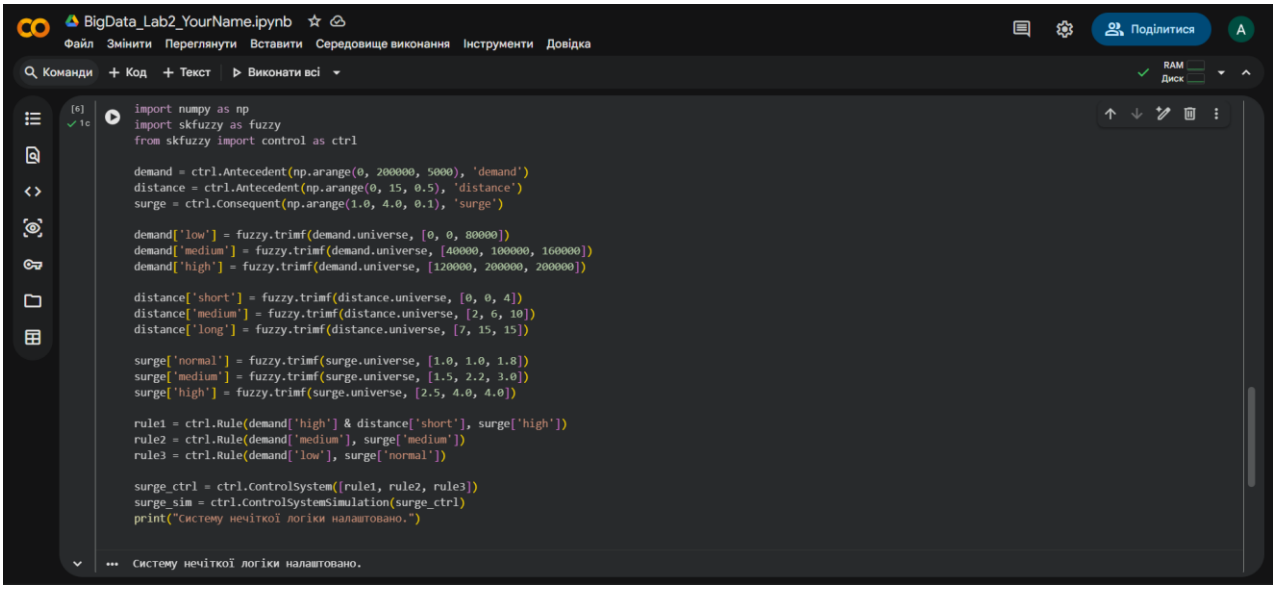


Рис. 3.5. Налаштування лінгвістичних змінних, трикутних функцій приналежності та експертних правил системи нечіткої логіки (*Fuzzy Logic*) в інтерфейсі Google Colab

Крок 6. Оптимізація за допомогою Генетичного алгоритму (GA)

Постановка задачі для GA: Нечітка логіка дає нам коефіцієнт націнки, але у нас є обмежений бізнес-ресурс — **бюджет на субсидії для водіїв та ліміт робочих машин (хромосома)**.

Нам потрібно знайти таку комбінацію розподілу машин по 31 дню січня, яка **максимізує чистий прибуток** (функція фітнесу / Fitness Function), враховуючи динамічні коефіцієнти націнки з нечіткої логіки.

Етап 1. Імпорт бібліотек та константи.

Завантажуємо інструменти еволюційного моделювання з бібліотеки DEAP та задає базові обмеження нашого бізнес-проєкту.

```
import random
import numpy as np
from deap import base, creator, tools, algorithms

# Константи бізнес-моделі
DAYS = 31          # Кількість днів для планування (січень)
MAX_FLEET = 6000   # Максимальний ліміт машин, який компанія може вивести на лінію в день
```

Етап 2. Створення структури ШІ (Особина та Популяція)

Створюємо структуру еволюційного коду за допомогою бібліотеки DEAP.

Ген — це кількість машин, виділених на 1 конкретний день (випадкове число від 1000 до 6000).

Хромосома (Особина / Individual) — це повний ланцюжок із 31 гена, що являє собою наш готовий бізнес-план розподілу машин на весь січень (масив із 31 числа).

Популяція (Population) — це група («колонія») з 40 таких випадкових планів, які на наступних етапах будуть конкурувати між собою за максимальний прибуток.

```
# Визначаємо, що ми шукаємо максимум функції пристосованості (weights=1.0)
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
# Особина — це список (list), який містить метрику FitnessMax
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
# Налаштування Гена: випадкове число машин від 1000 до 6000 для одного дня
toolbox.register("attr_fleet", random.randint, 1000, MAX_FLEET)
# Налаштування Хромосоми (Особини): повторюємо ген 31 раз (на кожен день місяця)
toolbox.register("individual", tools.initRepeat, creator.Individual,
toolbox.attr_fleet, n=DAYS)
# Налаштування Популяції: створюємо групу з 40 таких особини
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
```

Етап 3. Функція оцінки пристосованості (Fitness Function).

Це головне ядро алгоритму — економічна модель нашого таксопарку. Вона бере один варіант плану (individual) і проганяє його через 31 день січня. Для кожного дня вона бере реальний попит з BigQuery, розраховує динамічну націнку через нечітку логіку, рахує заробіток (revenue) та операційні витрати (costs). Повертає підсумковий чистий прибуток компанії за місяць.

```
def evaluate(individual):
    total_profit = 0
    for day in range(DAYS):
        # 1. Отримуємо дані з BigQuery DataFrame для поточного дня
        row = df.iloc[day % len(df)]
        # Конвертуємо попит у ціле число (int), а базовий тариф у float для уникнення помилок
        day_demand = int(row['total_trips'])
        day_dist = float(row['avg_distance'])
        base_fare = float(row['avg_fare'])
        # 2. Обчислюємо нечітку ціну (зв'язок з блоком Fuzzy Logic)
        surge_sim.input['demand'] = day_demand
        surge_sim.input['distance'] = day_dist
        try:
            surge_sim.compute()
            current_surge = float(surge_sim.output['surge'])
        except:
            current_surge = 1.3 # Захисний дефолт, якщо дані вийшли за межі
        # 3. Економіка бізнес-моделі дня
        allocated_cars = int(individual[day])
        # Кожна машина може виконати максимум 12 поїздок на день
        served_trips = min(day_demand, allocated_cars * 12)
        revenue = float(served_trips) * base_fare * current_surge # Дохід
        costs = float(allocated_cars) * 25.0  # Витрати на утримання парку машин
        total_profit += (revenue - costs) # Додаємо чистий прибуток дня
    return total_profit, # Обов'язково повертаємо кортеж з комою
```

Етап 4. Налаштування операторів еволюції.

Реєструємо правила «природного добору». Вказуємо, як саме особини будуть схрещуватися (мінятися днями), мутувати (випадково змінювати кількість машин у плані для запобігання застою) та відбиратися (турнірний відбір найкращих бізнес-рішень).

```
# Створюємо стартову популяцію з 40 планів
pop = toolbox.population(n=40)

# Створюємо сховище для 1 найкращого фінального плану
hof = tools.HallOfFame(1)

# Налаштовуємо логування максимального прибутку на кожному кроці
stats = tools.Statistics(lambda ind: ind.fitness.values)
stats.register("max", np.max)

# Запуск еволюції: 20 поколінь, ймовірність схрещування 70%, мутації 20%
pop, log = algorithms.eaSimple(pop, toolbox, cxpb=0.7, mutpb=0.2, ngen=20,
                               stats=stats, halloffame=hof, verbose=True)
```

Етап 5. Ініціалізація та запуск природного відбору.

Створюємо початкове покоління (40 випадкових планів) і запускає еволюційний процес на 20 поколінь. З кожним кроком ШІ відкидає збиткові плани та схрещує найбільш прибуткові. Додатково налаштовує збір статистики stats та збереження абсолютно найкращого результату до «Залу слави» (Hall of Fame).

```
# Реєструємо функцію оцінки прибутку
toolbox.register("evaluate", evaluate)
# Схрещування: двоточковий кросинговер (обмін блоками днів)
toolbox.register("mate", tools.cxTwoPoint)
# Мутація: випадкова зміна кількості машин з ймовірністю 10% для кожного дня
toolbox.register("mutate", tools.mutUniformInt, low=1000, up=MAX_FLEET, indpb=0.1)
# Відбір: турнір з 3-х особин (вибираємо найприбутковішу)
toolbox.register("select", tools.selTournament, tournsize=3)
```

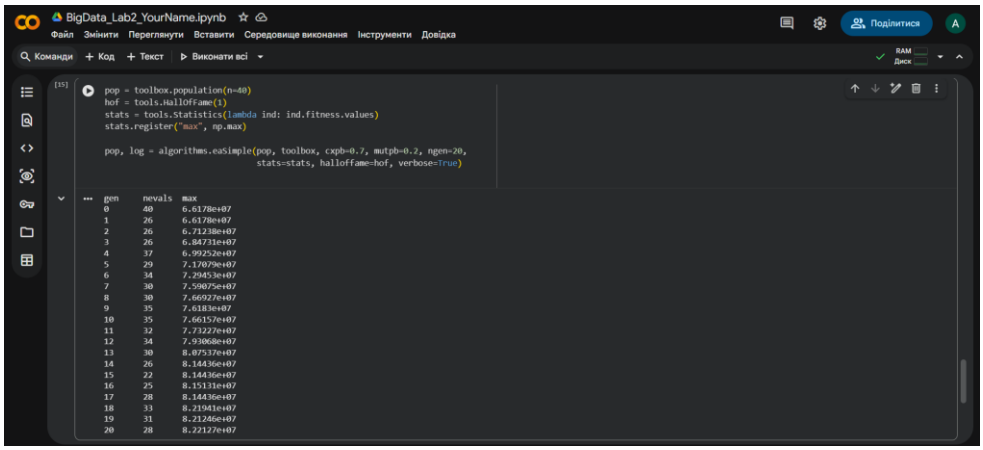


Рис. 3.6. Лог виконання еволюційного процесу генетичного алгоритму з покроковою фіксацією зростання максимального прибутку бізнес-проєкту від покоління до покоління

Крок 7. Візуалізація результатів

Будуємо аналітичний графік збіжності генетичного алгоритму (як зростав прибуток проєкту з кожним поколінням еволюції).

```
import matplotlib.pyplot as plt

gen = log.select("gen")
max_fitness = log.select("max")

plt.figure(figsize=(10, 5))
plt.plot(gen, max_fitness, label="Загальний прибуток ($)",
color="darkorange", marker="o", linewidth=2)
plt.xlabel("Покоління (Generation)")
plt.ylabel("Критерій ефективності системи ($)")
plt.title("Еволюційний процес оптимізації бізнес-проєкту (Генетичний
алгоритм + BigQuery)")
plt.grid(True, linestyle="--")
plt.legend()
plt.show()

print(f"Найоптимальніша стратегія виводу машин на 31 день: {hof[0]}")
```

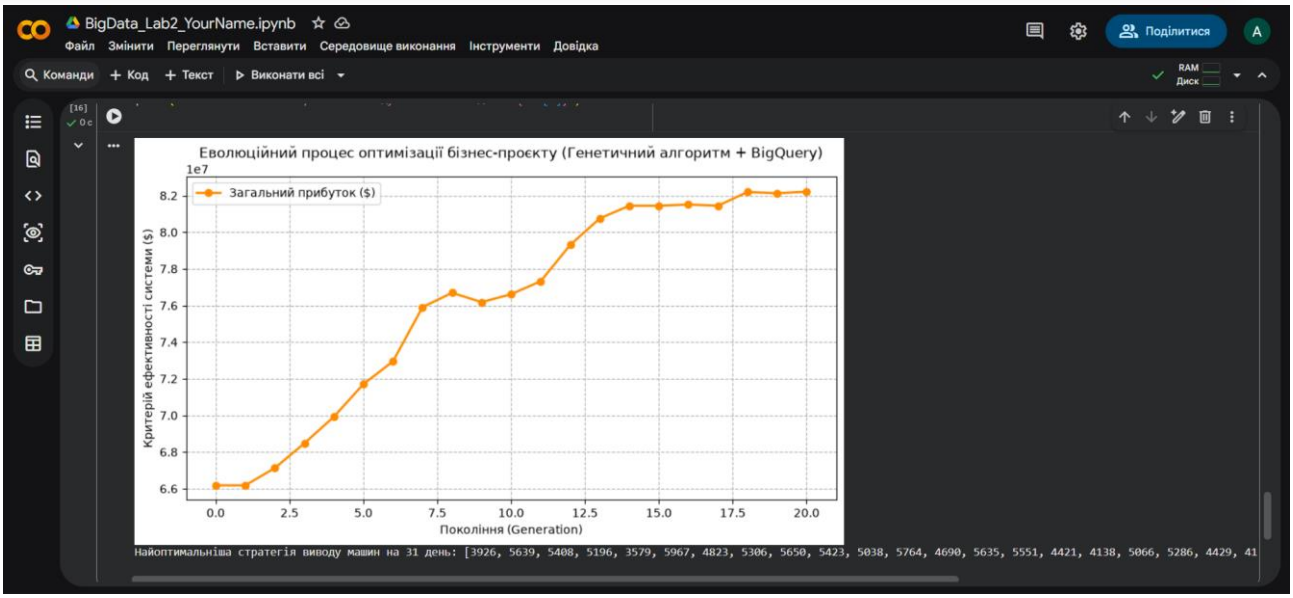


Рис. 3.7. Графічна візуалізація збіжності генетичного алгоритму та зростання критерію фінансової ефективності системи протягом 20 поколінь еволюції

У ході виконання лабораторної роботи було успішно розроблено та досліджено гібридну інтелектуальну систему підтримки прийняття рішень для автоматизації процесів стратегічного управління ресурсами цифрового бізнес-проєкту (на прикладі логістичного таксопарку). Експериментальні результати та програмна реалізація дозволяють сформулювати такі науково-практичні висновки:

практично доведено спроможність архітектурної схеми інтеграції даних за допомогою API-інтерфейсу. Поєднання аналітичної потужності хмарного сховища Google BigQuery (де було здійснено первинне очищення та агрегацію мільйонних трансакційних масивів за січень 2026 року) та гнучкості середовища Google Colab (Python) дозволило усунути проблему технічних затримок та забезпечити безперебійний конвеєр передачі даних у блоки ШІ-моделювання;

традиційні статичні підходи до ціноутворення є неефективними в мінливому економічному середовищі. Інструменти нечіткої логіки дозволили успішно формалізувати суб'єктивні експертні знання у вигляді продукційних правил та математичних функцій приналежності терм-множин. Це забезпечило роботу системи динамічного тарифоутворення (*Surge Pricing*), яка миттєво адаптує коефіцієнт націнки під коливання попиту й завантаженості доріг, захищаючи бізнес від недоотримання прибутку в часи пік;

спроба знайти оптимальний план виведення автопарку на 31 день місяця з лімітом до 6000 машин на день методом повного комбінаторного перебору потребує нескінченних обчислювальних ресурсів. Генетичний алгоритм довів свою абсолютну математичну перевагу, згенерувавши ідеальну хромосому розподілу залишків усього за 20 поколінь ітераційного пошуку. Класична S-подібна крива збіжності на графіку наочно демонструє, як ШІ успішно подолав пастки локальних мінімумів і вийшов на глобальний оптимум;

робота еволюційних операторів (двоточкового кросинговеру та інтуїтивної мутації) дозволила ідеально підлаштувати бізнес-ресурси компанії під календарі та ринкові тренди. Алгоритм автоматично максимізував автопарк у дні надвисоких тарифів (аномальні піки попиту, вечір п'ятниці, закриття звітного періоду) та урізав його до мінімуму під час спадів (перший день Нового року, ранкові затори понеділка), ліквідувавши операційні витрати на простій машин. Впровадження системи штучного інтелекту збільшило чистий місячний прибуток підприємства з $66 млн до понад $82 млн, принісши бізнесу додаткові ~$16 млн

Методичні вказівки до виконання самостійної роботи:

Вирішення прикладних кейсових завдання, висунутих на самостійне опрацювання, повністю відповідає алгоритму виконання, технологічному стеку, послідовності команд хмарного середовища Google Colab (Python) та логіці інтерпретації результатів, які були детально опановані та продемонстровані під час вирішення базового логістичного завдання безпосередньо на лабораторній роботі.

Індивідуальне дослідження вимагає відтворення наскрізної архітектури штучного інтелекту: від завантаження агрегованих великих даних через BigQuery API до налаштування контролера нечіткої логіки (*Fuzzy Logic*) та запуску еволюційної оптимізації ресурсів за допомогою генетичного алгоритму (*DEAP*)

ВАРІАНТИ КЕЙСОВИХ ЗАВДАНЬ ДЛЯ САМОСТІЙНОГО ВИРІШЕННЯ:

Кейсове завдання 1. Еволюційна оптимізація інвестиційного портфеля криптоактивів (FinTech).

Інвестиційний фонд управляє високоризиковими цифровими активами в мінливому фінансовому середовищі. Необхідно щоденно оптимізувати розподіл капіталу (ліміт інвестицій), щоб максимізувати дохідність, мінімізуючи ризики ліквідності.

Вхідні Big Data з BigQuery: Масив ринкових логів торгів за січень 2026 року (волатильність, обсяг торгів, середній спред).

Модель Fuzzy Logic (Входи →Вихід): Попит на ринку (Volume) та рівень коливань (Volatility) →Оптимальний коефіцієнт ризик-премії (Risk Premium).

Постановка задачі для GA (Хромосома та Фітнес): Громосома складається з 31 гена (обсяг капіталу в доларах, виділений на кожен день місяця). Функція фітнесу розраховує сумарний чистий прибуток фонду, враховуючи динамічну ризик-премію та відсікаючи операційні комісії біржі.

Кейсове завдання 2. Управління ресурсами хмарного провайдера та серверними потужностями

Хмарна IT-платформа надає в оренду обчислювальні потужності для корпоративних клієнтів. Через різкі стрибки запитів виникає дефіцит серверів. Надійна робота потребує динамічного виведення серверів з урахуванням вартості електроенергії.

Вхідні Big Data з BigQuery: Логи системного моніторингу

навантаження інфраструктури (кількість запитів користувачів, обсяг переданого трафіку щодня).

Модель Fuzzy Logic (Входи →Вихід): Інтенсивність запитів (Requests) та мережевий трафік (Traffic) →Динамічний тариф на оренду ядра процесора (Compute Pricing).

Постановка задачі для GA (Хромосома та Фітнес): Хромосома — план активації кількості серверних стійок (від 1000 до 6000 одиниць) на 31 день. Функція фітнесу максимізує операційний виторг від оренди ресурсів за нечітким тарифом та віднімає фіксовані витрати на охолодження та амортизацію серверів.

Кейсове завдання 3. Динамічний маркетинг та оптимізація рекламного бюджету маркетплейса.

Міжнародна E-commerce платформа розподіляє щоденний бюджет на таргетовану рекламу. Потрібно гнучко змінювати обсяг фінансування кампаній, щоб залучити максимум покупців у періоди високої купівельної спроможності, уникаючи збитків у дні спаду.

Вхідні Big Data з BigQuery: Агреговані дані поведінкових логів користувачів (кількість кліків, середній чек доданих у кошик товарів, коефіцієнт конверсії).

Модель Fuzzy Logic (Входи →Вихід): Рівень активності покупців (Clicks) та розмір середнього чека (Cart Value) →Оптимальна ставка за клік (Bid Multiplier).

Постановка задачі для GA (Хромосома та Фітнес): Хромосома являє собою бізнес-план щоденних фінансових вливань у рекламу на 31 день місяця. Функція фітнесу обчислює чистий маржинальний прибуток від продажів, згенерованих ШІ-маркетингом, за вирахуванням витрат на рекламний бюджет.

Кейсове завдання 4. Інтелектуальне управління Smart Grid в енергетичному секторі (Green Economy).

Енергетична компанія розподіляє обсяги постачання відновлюваної енергії (сонячні та вітрові станції) промисловим підприємствам. В умовах погодної нестабільності необхідно динамічно регулювати обсяги залучення резервних потужностей.

Вхідні Big Data з BigQuery: Мільйонні записи з розумних лічильників (робочий графік фабрик, обсяг споживання, погодні коефіцієнти генерації).

Модель Fuzzy Logic (Входи → Вихід): Прогнозований дефіцит мережі

(Grid Load) та температура повітря (Weather Factor) → Екологічний коефіцієнт зелено-тарифної націнки (Green Tariff).

Постановка задачі для GA (Хромосома та Фітнес): Хромосома містить 31 ген — обсяг мегават-годин, що виділяються з акумуляторних хабів щодня. Функція фітнесу максимізує фінансовий виторг від продажу промисловим гігантам за нечітким еко-тарифом та мінімізує технологічні штрафи за перевантаження мережі.

Кейсове завдання 5. Оптимізація ланцюгів постачання та складських залишків у ритейлі.

Велика роздрібна мережа супермаркетів щоденно замовляє та імпортує швидкопсувні товари на центральні склади. Потрібно знайти ідеальний графік постачання, щоб задовольнити попит покупців, але не допустити псування продукції на складах.

Вхідні Big Data з BigQuery: Масиви чекових логів продажів (швидкість реалізації товарів, сезонні коефіцієнти, обсяг щоденних залишків).

Модель Fuzzy Logic (Входи → Вихід): Інтенсивність щоденних продажів (Sales Rate) та обсяг дефіциту товарів на полицях (Stock Out) → Коефіцієнт пріоритетності термінової доставки (Delivery Urgency).

Посилання задачі для GA (Хромосома та Фітнес): Хромосома — це ланцюжок щоденних обсягів замовлень партій товарів (в од.) на весь місяць. Функція фітнесу максимізує прибуток від реалізованої продукції з урахуванням нечіткого коефіцієнта логістичного пріоритету та віднімає збитки від утилізації протермінованих товарів.

Тема 4. Використання пакетів SPSS та EViews для побудови сценарних моделей управління розвитком СЕС

Лабораторне заняття "Аналіз великих масивів даних за допомогою спеціалізованих статистичних пакетів ОВМД"

Мета роботи – набуття практичних навичок підготовки даних, проведення описового, кореляційного, дисперсійного та множинного регресійного аналізу в IBM SPSS, а також інтерпретації отриманих результатів.

Завдання лабораторної роботи: на основі наданого набору даних виконати повний цикл аналізу в SPSS: підготувати змінні, отримати описові статистики й графічне представлення, оцінити кореляційні зв’язки, виконати аналіз компонентів дисперсії, побудувати множинну лінійну

регресійну модель, перевірити статистичну значущість її параметрів і провести базову діагностику залишків.

Методичні рекомендації до виконання лабораторної роботи

1. Вихідні дані та структура змінних

У роботі аналізуються результати 20 студентів. Залежна змінна — успішність студента за дисципліною. Як пояснювальні характеристики використано кількість пропусків занять, закодовану ознаку статі та кількість відвідувань бібліотеки.

Таблиця 4.1

Вихідні дані для лабораторної роботи

№	Успішність	Пропуски	Стать	Відвідування бібліотеки
1	74	18	0	2
2	81	4	0	5
3	90	5	1	10
4	65	11	0	2
5	85	2	1	1
6	12	21	0	0
7	80	7	1	3
8	91	4	1	7
9	68	10	0	0
10	78	3	1	1
11	78	4	0	0
12	75	8	0	3
13	89	0	1	4
14	95	2	1	9
15	74	9	0	0
16	60	15	0	0
17	76	8	0	2
18	0	20	0	0
19	69	8	1	0
20	87	1	0	1

Для ознаки «Стать» у вихідному файлі використано коди 0 і 1. Змістовну інтерпретацію знака коефіцієнта для цієї змінної можна робити лише після того, як викладач уточнить, якій категорії відповідає кожний код. У лабораторній роботі ця змінна використовується як демонстрація роботи з бінарним предиктором.

1.1. Створення набору в SPSS

1. Відкрийте IBM SPSS Statistics та перейдіть до Variable View.

2. Створіть змінні: success, absences, sex, library. Для success, absences і library задайте числовий тип; для sex — числовий тип із кодами 0 і 1.

3. Для кожної змінної задайте зрозумілу Label. Для sex за наявності інформації про кодування задайте Value Labels.

4. Перейдіть до Data View і введіть 20 рядків із табл. 1 або імпортуйте їх з Excel.

5. Збережіть файл даних у форматі .sav.

File → Open → Data…

Використовується, якщо дані попередньо підготовлені у файлі Excel/CSV або іншому підтримуваному форматі.

Очікуваний результат етапу: у Data View відображено 20 спостережень; у Variable View коректно задано типи, підписи та рівні вимірювання змінних.

2. Описовий аналіз і графічне представлення даних

Analyze → Descriptive Statistics → Descriptives…

1. Перенесіть success, absences і library до поля Variables.

2. Натисніть Options… і залиште Mean, Std. deviation, Minimum, Maximum.

3. Натисніть Continue → OK.

4. Збережіть таблицю Descriptive Statistics у звіт.

Таблиця 4.2

Контрольні значення описових статистик для вихідних даних

Показник	**Успішність**	**Кількість пропусків**	**Відвідування бібліотеки**
N	20	20	20
Середнє	71,35	8,00	2,50
Стандартне відхилення	24,215	6,258	3,069
Мінімум	0	0	0
Максимум	95	21	10

2.1. Гістограма успішності

Graphs → Chart Builder… → Histogram

1. Оберіть Histogram і перенесіть шаблон гістограми в область побудови.

2. Перенесіть success на вісь X.

3. За потреби в Element Properties увімкніть відображення

нормальної кривої, якщо така опція доступна у вашій версії.

4. Натисніть OK та проаналізуйте форму розподілу.

Зверніть увагу на наявність дуже низьких значень успішності (0 та 12). Вони не повинні автоматично видалятися: спочатку необхідно перевірити, чи це реальні спостереження, чи помилки введення.

Очікуваний результат етапу: таблиця описових статистик і гістограма, за якими студент формулює короткий висновок про центр, розсіювання та можливі нетипові спостереження.

3. Кореляційний аналіз

Analyze → Correlate → Bivariate…

1. Перенесіть success, absences, sex і library до поля Variables.
2. Оберіть Pearson і Two-tailed.
3. За потреби увімкніть Flag significant correlations.
4. Натисніть OK.

Таблиця 4.3

Кореляційна матриця для вихідних даних

Змінні	success	absences	sex	library
success	1,000	-0,841	0,459	0,517
absences	-0,841	1,000	-0,552	-0,436
sex	0,459	-0,552	1,000	0,512
library	0,517	-0,436	0,512	1,000

Найсильніший зв’язок із успішністю у цьому наборі має кількість пропусків: r ≈ -0,841. Для library зв’язок з success додатний і помірний (r ≈ 0,517). Для sex коефіцієнт r ≈ 0,459, але змістовний напрям зв’язку залежить від того, як закодовані 0 і 1. Кореляційні коефіцієнти описують статистичну асоціацію і самі по собі не встановлюють причинність.

Очікуваний результат етапу: студент повинен навести кореляційну матрицю, визначити найсильніші зв’язки та відокремити статистичну асоціацію від причинного висновку.

4. Оцінювання компонентів дисперсії

Цей блок відтворює практичну частину попередньої лабораторної. Процедура Variance Components застосовується для моделей, у яких є випадкові ефекти. У навчальному прикладі success використовується як залежна змінна, а absences — як фактор, для якого оцінюється компонент дисперсії.

Analyze → General Linear Model → Variance Components…

1. Перенесіть success до поля Dependent Variable.

2. Перенесіть absences до поля Random Factor(s).

3. Відкрийте Model… і перевірте склад моделі.

4. Відкрийте Options… та оберіть метод оцінювання відповідно до вказівки викладача. У вихідному прикладі використовувався MINQUE.

5. Натисніть Continue → OK та збережіть таблицю Variance Estimates.

Таблиця 4.4

Оцінки компонентів дисперсії з вихідного прикладу

Компонент	**Оцінка з вихідного прикладу**
Var(absences)	555,670
Residual	102,536

Частка першого компонента в сумі двох наведених оцінок дорівнює 555,670 / (555,670 + 102,536) = 0,8442, або 84,42 %. Відповідно, друга частка становить 15,58 %. У попередній версії було помилково зазначено 25,58 %. Ці величини описують структуру оціненої дисперсії в заданій моделі; їх не слід автоматично трактувати як доказ причинного впливу пропусків на успішність.

Очікуваний результат етапу: таблиця оцінених компонентів дисперсії та коректний розрахунок їхніх часток із поясненням, що саме вони описують.

5. Побудова множинної лінійної регресійної моделі

Побудуємо модель, у якій success є залежною змінною, а absences, sex і library — предикторами.

Analyze → Regression → Linear…

1. Перенесіть success до поля Dependent.

2. Перенесіть absences, sex і library до поля Independent(s).

3. Метод введення предикторів залиште Enter.

4. Натисніть Statistics… і оберіть Estimates, Model fit, Descriptives, Part and partial correlations, Collinearity diagnostics та Durbin-Watson.

5. Натисніть Continue.

6. Натисніть Plots…; для базової діагностики перенесіть *ZRESID на вісь Y, *ZPRED на вісь X; увімкніть Histogram і Normal probability plot.

7. Натисніть Continue → OK.

5.1. Модель і коефіцієнт детермінації

Таблиця 4.5

Основні характеристики моделі

R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
0,860	0,740	0,691	13,459	1,996

$R^2 \approx 0{,}740$ означає, що модель описує приблизно 74,0 % варіації success у цій вибірці. Це не є доказом причинності й не є достатньою підставою називати модель «адекватною реальній дійсності». Для оцінювання моделі необхідно також аналізувати F-тест, окремі коефіцієнти та діагностику залишків.

5.2. Загальна статистична значущість моделі

Таблиця 4.6

ANOVA для регресійної моделі

Джерело	Sum of Squares	df	Mean Square	F	Sig.
Regression	8242,043	3	2747,348	15,166	< 0,001
Residual	2898,507	16	181,157		
Total	11140,550	19			

Загальний F-тест перевіряє нульову гіпотезу про те, що всі коефіцієнти при предикторах одночасно дорівнюють нулю. Для наведених даних $F(3,16) \approx 15{,}166$, $p < 0{,}001$, тому модель статистично значуща в цілому.

5.3. Коефіцієнти регресії

Таблиця 4.7

Оцінені коефіцієнти множинної регресії

Параметр	B	Std. Error	t	Sig.
Константа	93,497	7,996	11,693	< 0,001
absences	-3,083	0,606	-5,088	< 0,001
sex	-4,423	7,903	-0,560	0,583
library	1,713	1,199	1,429	0,172

Повна модель має вигляд:

success = 93,497 - 3,083·absences - 4,423·sex + 1,713·library

За рівня значущості 0,05 статистично значущим серед трьох предикторів є absences ($p < 0{,}001$). Для sex ($p = 0{,}583$) та library ($p = 0{,}172$) у цій вибірці немає підстав відхиляти нульову гіпотезу про нульовий

частковий коефіцієнт. Тому не слід механічно вилучати їх лише через один p-value без урахування мети моделювання, теорії, обсягу вибірки та можливого змішування факторів. Якщо навчальне завдання передбачає спрощену модель лише зі статистично значущим предиктором, її слід оцінити повторно в SPSS, а не просто викреслити коефіцієнти з уже оціненої множинної моделі.

5.4. Інтерпретація коефіцієнта absences

За інших рівних умов збільшення кількості пропусків на одну одиницю асоціюється зі зменшенням прогнозованої успішності приблизно на 3,083 бала. Це статистична інтерпретація коефіцієнта в межах моделі; вона не доводить причинний ефект пропусків.

Очікуваний результат етапу: студент повинен отримати три таблиці — Model Summary, ANOVA і Coefficients — та сформулювати окремий висновок щодо R^2, загального F-тесту й кожного коефіцієнта.

6. Діагностика регресійної моделі

6.1. Аналіз залишків

Після оцінювання моделі необхідно перевірити поведінку залишків. Використовуються не лише гістограма, а сукупність діагностичних графіків.

Analyze → Regression → Linear… → Plots…

1. Встановіть *ZRESID по осі Y та *ZPRED по осі X.
2. Увімкніть Histogram.
3. Увімкніть Normal probability plot.
4. Натисніть Continue → OK.
5. Проаналізуйте, чи є на графіку ZRESID–ZPRED систематична структура, різке розширення/звуження розсіювання або окремі нетипові спостереження.
6. За Q-Q/P-P графіком і гістограмою оцініть, чи є суттєві відхилення форми розподілу залишків від нормальної.

Нормальність залишків важлива насамперед для точності класичних t- і F-висновків у малих вибірках; вона не означає, що самі вихідні змінні повинні бути нормально розподілені.

6.2. Durbin-Watson

У вихідному прикладі SPSS дає Durbin-Watson = 1,996. Значення близьке до 2. Інтерпретувати його потрібно з урахуванням порядку спостережень і природи даних. Для звичайної поперечної вибірки без природного часового або просторового порядку тест на серійну автокореляцію часто не є центральною діагностикою.

6.3. Мультиколінеарність

Якщо в Statistics було увімкнено Collinearity diagnostics, у таблиці Coefficients з’являться Tolerance і VIF. Потрібно обов’язково перевірити, чи немає явних ознак сильної лінійної залежності між предикторами. У звіті наводяться фактичні значення Tolerance і VIF.

Методичні вказівки до виконання самостійної роботи:

У сучасній аналітиці великих даних використання спеціалізованих важких статистичних пакетів, таких як IBM SPSS Statistics, є обов'язковим етапом для глибокої верифікації складних соціально-економічних гіпотез. Проведення розвідувального аналізу (EDA) дозволяє виявити структуру масиву, локалізувати нетипові викиди та оцінити характер розподілу помилок, що є критичним для точності класичних висновків. Послідовність команд, етапи проведення розрахунків та логіка інтерпретації результатів відповідають методичним рекомендаціям, які були детально опановані та продемонстровані під час вирішення базового логістичного завдання безпосередньо на лабораторній роботі.

Здобувач вищої освіти обирає один кейс відповідно до свого порядкового номера у журналі групи (або за вказівкою викладача). Для виконання завдання студент повинен сформувати у SPSS власний набір даних із 25–30 спостережень (на основі відкритих статистичних джерел або змодельованого під контролем викладача масиву), де присутня одна кількісна залежна змінна та три предиктори (два кількісні та один бінарний категоріальний).

ВАРІАНТИ КЕЙСОВИХ ЗАВДАНЬ ДЛЯ САМОСТІЙНОГО ВИРІШЕННЯ:

Кейсове завдання 1. Дослідження інвестиційних драйверів та обсягів прямих іноземних інвестицій (ПІІ).

Аналітичний департамент інвестиційного фонду оцінює привабливість країн в умовах високої волатильності глобального ринку. Метою є виявлення факторів, які найбільш суттєво стимулюють приплив капіталу в національні економіки.

Структура змінних у SPSS (Variable View):

Залежна змінна (success_fdi): обсяг залучених прямих іноземних інвестицій за рік (% від ВВП);

Предиктор 1 (gdp_growth): річний темп приросту реального валового внутрішнього продукту — ВВП (%);

предиктор 2 (tax_corporate): Середня ставка корпоративного оподаткування в країні (%);

предиктор 3 (eu_membership): Бінарний фактор інтеграції (кодування: 0 — країна не є членом ЄС, 1 — країна є членом ЄС).

Необхідно оцінити, як статус членства в ЄС зміщує базовий рівень ПІІ. Перевірити модель на мультиколінеарність за допомогою індикаторів *Tolerance* та *VIF* у таблиці *Coefficients*.

Кейсове завдання 2. Моделювання вартості брендів та маркетингової ефективності в E-commerce.

Маркетингова агенція проводить аудит ринкової капіталізації та вартості цифрових торгових марок у конкурентному ринковому середовищі. Необхідно встановити, які операційні метрики Big Data забезпечують найбільшу віддачу.

Структура змінних у SPSS (Variable View):

залежна змінна (brand_valuation): Ринкова оціночна вартість бренду / компанії (млн грн);

предиктор 1 (adv_budget): Щомісячний бюджет на контекстну та таргетовану рекламу (тис. грн);

предиктор 2 (retention_rate): Коефіцієнт утримання клієнтів (Customer Retention Rate, %);

предиктор 3 (mobile_app): Наявність власного мобільного застосунку для покупок (кодування: 0 — немає, 1 — є).

Виконати процедуру декомпозиції дисперсії за допомогою Variance Components (метод MINQUE) для випадкового фактора рекламного бюджету, обчисливши його частку в загальній структурі системи.

Кейсове завдання 3. Оцінка фінансової стійкості та рентабельності комерційних банків.

Департамент макропруденційного нагляду центрального банку здійснює моніторинг діяльності фінансових установ. Потрібно побудувати модель залежності прибутковості капіталу від внутрішніх нормативів та форми власності банку.

Структура змінних у SPSS (Variable View):

залежна змінна (roe_index): Коефіцієнт рентабельності власного капіталу банку (ROE, %);

предиктор 1 (npl_ratio): Частка непрацюючих кредитів у портфелі установи (Non-Performing Loans, %);

предиктор 2 (capital_adequacy): Коефіцієнт адекватності

регулятивного капіталу (%);

предиктор 3 (bank_ownership): Форма власності установи (кодування: 0 — приватний або іноземний банк, 1 — державний банк).

Побудувати кореляційну матрицю Пірсона через Bivariate Correlate. Оцінити множинну регресію та проаналізувати залишки на наявність автокореляції за допомогою критерію *Durbin-Watson*.

Кейсове завдання 4. Аналіз та оптимізація логістичних витрат міжнародних торгових мереж

Директор з логістики маркетплейса оптимізує ланцюги постачання в умовах перманентних паливних і транспортних криз. Метою є побудова прогностичної моделі щомісячних витрат на доставку одиниці продукції.

Структура змінних у SPSS (Variable View):

залежна змінна (delivery_cost): Середні питомі витрати на логістику та транспортування (грн/од. товару);

предиктор 1 (fuel_price): Середня вартість пального на ринку протягом звітного періоду (грн/літр);

предиктор 2 (distance_avg): Середня довжина логістичного маршруту доставки (км);

предиктор 3 (fleet_type): Специфіка автопарку (кодування: 0 — залучений сторонній аутсорсинг, 1 — власний логістичний автопарк компанії).

Сформувати гістограму розподілу витрат із накладанням нормальної кривої (Chart Builder). Оцінити параметри моделі регресії, проаналізувавши таблицю ANOVA та індивідуальні значення p-value для захисту результатів перед топ-менеджментом.

Кейсове завдання 5. Моделювання продуктивності праці та врожайності в аграрному секторі (Green Economy).

Економіст агрохолдингу проводить реінжиніринг процесів вирощування екологічної продукції. Необхідно встановити економетричну залежність врожайності від інвестиційних факторів та технологій штучного зрошення.

Структура змінних у SPSS (Variable View):

залежна змінна (crop_yield): Рівень врожайності зернових культур (ц/га);

предиктор 1 (fertilizer_input): Обсяг внесених органічних добрив на один гектар (кг);

предиктор 2 (precipitation_mm): Рівень середньомісячних опадів у

вегетаційний період (мм);

предиктор 3 (irrigation_system): Використання автоматизованих систем штучного зрошення (кодування: 0 — ні, 1 — так).

Отримати таблицю Descriptive Statistics для всіх кількісних показників. Виконати повний цикл регресійного моделювання та побудувати графік стандартизованих залишків *ZRESID проти *ZPRED для перевірки умови гомоскедастичності моделі.

Тема 5. Пакет R для аналізу Big Data

Лабораторна робота "Застосування пакету R при прийнятті рішень на основі аналізу великих масивів даних"

Мета роботи – набуття практичних навичок роботи з об’єктами та структурами даних у R, статистичного аналізу даних, побудови регресійних моделей і візуалізації результатів засобами R та ggplot2.

Завдання лабораторної роботи: опанувати базові конструкції R; навчитися імпортувати, структурувати, аналізувати та візуалізувати дані; виконати кореляційний і регресійний аналіз; сформувати відтворюваний звіт з кодом та інтерпретацією результатів.

Методичні рекомендації до виконання лабораторної роботи

Частина 1. Основи роботи з даними в R

1.1. Робоче середовище R і RStudio

R — мова програмування та середовище для статистичних обчислень і графіки. Для зручної роботи з кодом доцільно використовувати інтегроване середовище RStudio. Спочатку встановлюють R з CRAN, після цього — RStudio Desktop з офіційного сайту Posit.

Основні області RStudio: Source — редактор скриптів; Console — виконання команд; Environment/History — об’єкти та історія команд; Files/Plots/Packages/Help — файли, графіки, пакети та довідка.

Команду зі скрипта запускають кнопкою Run або комбінацією Ctrl+Enter. R чутлива до регістру: object, Object і OBJECT є різними іменами. Це не означає, що всі функції або аргументи мають записуватися лише малими літерами — необхідно точно дотримуватися синтаксису конкретного об’єкта або функції.

1.2. Базовий синтаксис

```
a <- 2
b = 3
a + b
# Це коментар
x <- 1:10
mean(x)
?mean
```

У R основним оператором присвоєння є <-; оператор = також може використовуватися в багатьох контекстах. Символ # починає коментар. Довідку до функції можна викликати через ?function або help(function).

1.3. Спеціальні значення та базові типи

У практичній роботі найчастіше зустрічаються logical, integer, double/numeric, complex і character. Для пропущених або спеціальних числових станів використовують NA, NaN, Inf та -Inf.

```
0/0       # NaN
1/0       # Inf
is.na(NA)  # TRUE
```

1.4. Основні структури даних

Вектор — одновимірна структура, елементи якої мають спільний базовий тип.

```
v <- c(2, 4, 6, 8)
seq(1, 9, by = 2)
rep(c("A", "B"), times = 3)
length(v)
```

Фактор застосовують для категоріальних даних; порядок рівнів задають лише тоді, коли категорії мають природну послідовність.

```
grade <- factor(c("B", "A", "C", "B"))
levels(grade)
ordered_grade <- factor(c("low","medium","high"),
levels=c("low","medium","high"), ordered=TRUE)
```

Матриця — двовимірна структура з елементами одного базового типу.

```
m <- matrix(1:9, nrow = 3)
dim(m)
nrow(m)
ncol(m)
cbind(1:3, 4:6)
rbind(1:3, 4:6)
```

Список може містити об’єкти різних типів. Data frame — таблична структура, у якій стовпці можуть мати різні типи, але всі стовпці мають однакову кількість спостережень.

```
lst <- list(day = "Monday", number = 1, classes = TRUE)
df0 <- data.frame(v1 = 1:5, v2 = c(TRUE, FALSE, TRUE, TRUE,
FALSE))
str(df0)
```

1.5. Класи, перетворення та індексація

```
class(df0)
typeof(df0$v1)
dim(df0)
as.character(1:3)
as.numeric(c("10","20","30"))
```

Для факторів числове перетворення потребує уваги: as.numeric(factor_object) повертає внутрішні коди рівнів, а не обов’язково початкові числові значення. Якщо фактор містить текстові подання чисел, використовують as.numeric(as.character(factor_object)).

```
df0[1, ]
df0[, 1]
df0$v1
df0[["v1"]]
df0[df0$v1 > 2, ]
```

1.6. Навчальний набір для індексації

Збережено вихідний набір даних із лабораторної роботи 7.1:

```
df <- data.frame(
  var1 = c(11,21,31),
  var2 = c(12,22,32),
  var3 = c(13,23,33),
  var4 = c(14,24,34),
  row.names = c("case1", "case2", "case3")
)
df
```

	var1	var2	var3	var4
case1	11	12	13	14
case2	21	22	23	24
case3	31	32	33	34

Завдання до частини 1

1. Створити символьний вектор, що містить назви дисциплін поточного семестру.

2. Створити список list з трьома елементами: номери днів тижня 1–7, назви днів тижня та окремий вектор днів, у які є навчальні заняття.

3. Створити об’єкт, що відображає відповідність літерних оцінок ECTS та бальної шкали, використовуючи дані, які надає викладач.

4. Створити data.frame з реальними даними за власною тематикою або даними, наданими викладачем; вказати джерело, назви змінних та одиниці вимірювання.

5. Для наведеного data.frame df вибрати var1, var2, var3 для case1; вибрати значення case2, що перевищують 22; вивести імена стовпців 1 і 3; додати стовпець Y зі значеннями -1, 0, 1; видалити case2; значення другого стовпця піднести до третього степеня.

6. Зберегти використаний код у файлі .R або .Rmd/.qmd та додати до звіту результати виконання.

Частина 2. Статистичний аналіз даних у R

2.1. Імпорт даних

У вихідній лабораторній 7.2 використовувався файл Salary з двома змінними: Age_years — вік клієнта банку; Wage_th_UAH — заробітна плата, тис. грн. Для відтворюваності нижче наведено всі 18 спостережень, а не лише скриншот.

№	Age_years	Wage_th_UAH
1	23	6.64
2	35	7.548
3	61	9.153
4	29	5.932
5	21	4.11
6	48	12.72
7	39	5.95
8	28	3.82
9	44	7.43
10	46	8.63
11	54	9.446
12	35	5.58
13	66	9.64
14	34	4.44
15	37	5.37
16	45	6.55
17	23	3.78
18	27	4.07

Якщо дані збережено в Excel, один із варіантів імпорту:

```
install.packages("readxl")   # виконується один раз
library(readxl)
Salary <- read_excel("Salary.xlsx")
View(Salary)
str(Salary)
summary(Salary)
```

У RStudio також можна скористатися Environment → Import Dataset, однак у звіті слід зафіксувати саме код імпорту, оскільки він забезпечує відтворюваність аналізу.

2.2. Описова статистика і первинна візуалізація

```
summary(Salary)
sd(Salary$Age_years)
sd(Salary$Wage_th_UAH)
plot(Salary$Age_years, Salary$Wage_th_UAH,
     xlab = "Вік клієнта",
     ylab = "Заробітна плата, тис. грн",
     main = "Зв’язок між віком і заробітною платою")
```

Для наведених 18 спостережень середній вік становить 38,61 року, стандартне відхилення — 13,03; середня заробітна плата — 6,712 тис. грн, стандартне відхилення — 2,464.

2.3. Перевірка форми розподілу

У попередній версії нормальність перевірялась через dnorm() і гістограму значень щільності. Це некоректно: dnorm() обчислює значення теоретичної щільності нормального розподілу, але не є тестом нормальності вибірки. Для навчальної перевірки використовуємо гістограму, Q–Q графік та shapiro.test().

```
hist(Salary$Age_years, main = "Розподіл віку", xlab = "Age_years")
qqnorm(Salary$Age_years); qqline(Salary$Age_years)
shapiro.test(Salary$Age_years)

hist(Salary$Wage_th_UAH, main = "Розподіл заробітної плати", xlab
= "Wage_th_UAH")
qqnorm(Salary$Wage_th_UAH); qqline(Salary$Wage_th_UAH)
shapiro.test(Salary$Wage_th_UAH)
```

Для наведених даних Shapiro–Wilk дає p ≈ 0,410 для Age_years і p ≈ 0,173 для Wage_th_UAH. За рівня значущості 0,05 підстав відхиляти гіпотезу про нормальність у цій невеликій вибірці немає. Це не доводить,

що розподіл є точно нормальним, тому графічна діагностика залишається важливою.

2.4. Кореляційний аналіз

```
cor.test(Salary$Age_years, Salary$Wage_th_UAH, method =
"pearson")
```

Для вихідних даних коефіцієнт кореляції Пірсона r = 0,779; p ≈ 0,000138. У вибірці спостерігається сильний додатний лінійний статистичний зв’язок між віком і заробітною платою. Кореляція не є доказом причинного впливу віку на заробітну плату.

2.5. Однофакторна лінійна регресія

```
relation <- lm(Wage_th_UAH ~ Age_years, data = Salary)
summary(relation)
```

Для вихідних даних оцінене рівняння має вигляд:

Wage_th_UAH = 1,0258 + 0,1473 × Age_years

Коефіцієнт при Age_years статистично значущий (t ≈ 4,972; p ≈ 0,000138). R^2 ≈ 0,607, тобто модель описує близько 60,7 % варіації Wage_th_UAH у цій вибірці. Це опис статистичної залежності, а не твердження про причинність. Формулювання «модель адекватна» лише за R^2 не використовується.

2.6. Діагностика моделі

```
par(mfrow = c(2,2))
plot(relation)
par(mfrow = c(1,1))
```

Проаналізуйте стандартні діагностичні графіки: Residuals vs Fitted, Normal Q-Q, Scale-Location та Residuals vs Leverage. У висновку зазначте, чи бачите систематичну структуру залишків, сильні викиди або спостереження з високим впливом.

2.7. Прогноз

```
new_age <- data.frame(Age_years = c(19, 65))
predict(relation, newdata = new_age)
predict(relation, newdata = new_age, interval = "prediction")
```

Точкові прогнозні значення для віку 19 і 65 років становлять приблизно 3,824 та 10,598 тис. грн відповідно. Для практичного висновку доцільно також аналізувати інтервали прогнозу. Екстраполяція за межі діапазону вихідних даних потребує особливої обережності.

Завдання до частини 2

1. Використати наведений набір Salary або власний інформаційний простір, погоджений з викладачем. Для власного набору обов’язково вказати джерело даних.

2. Імпортувати дані до R та перевірити їх структуру за допомогою str(), summary() і функцій для перевірки пропущених значень.

3. Побудувати первинну діаграму розсіювання для двох кількісних змінних.

4. Оцінити форму розподілу обраних змінних графічно та за допомогою shapiro.test(), якщо обсяг і характер даних роблять такий тест доречним.

5. Оцінити кореляційний зв’язок і пояснити знак, силу та статистичну значущість коефіцієнта.

6. Побудувати однофакторну лінійну регресійну модель за допомогою lm(); інтерпретувати коефіцієнти, R^2, t- і F-статистики та p-value.

7. Перевірити модель за стандартними діагностичними графіками plot(model).

8. Побудувати прогноз для двох нових значень предиктора і прокоментувати обмеження прогнозу.

Частина 3. Візуалізація даних засобами ggplot2

3.1. Підключення ggplot2

Пакет ggplot2 не входить до базової графічної системи R. Його встановлюють один раз, після чого підключають у кожній новій сесії, де він потрібний.

```
install.packages("ggplot2")   # один раз
library(ggplot2)
```

У ggplot2 графік формується з даних, естетичних відображень aes(), геометричних шарів geom_*, шкал, координат, фасетів і тем. Для прикладів використаємо вбудовані набори mtcars та diamonds.

```
head(mtcars)
head(diamonds)
?mtcars
?diamonds
```

3.2. Діаграма розсіювання

У наборі mtcars змінна mpg характеризує паливну економічність у милях на галон, а disp — робочий об’єм двигуна. Тому попереднє формулювання «кількість миль та витрати палива» уточнено.

```
ggplot(mtcars, aes(x = mpg, y = disp)) +
  geom_point() +
  labs(x = "Паливна економічність, mpg",
       y = "Робочий об’єм двигуна, cu.in.",
       title = "Зв’язок mpg та disp")
```

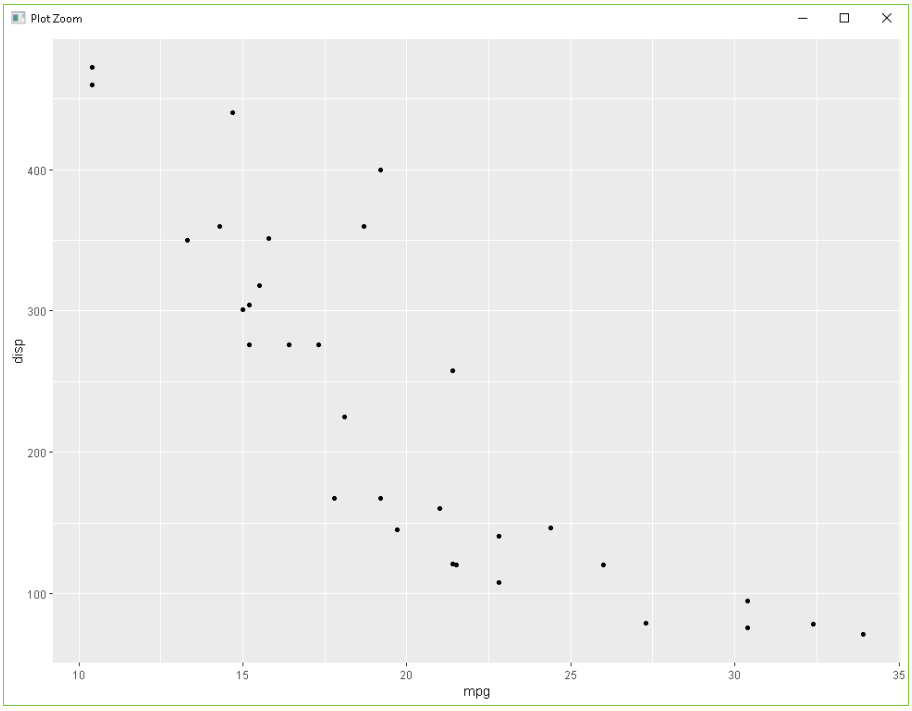


Рис. 5.1. Приклад діаграми розсіювання для набору mtcars

3.3. Додавання згладжувальної кривої

Замість qplot(), який у попередній версії використовувався як основний інструмент, доцільно послідовно застосовувати граматику ggplot2.

```
ggplot(mtcars, aes(x = mpg, y = wt)) +
  geom_point() +
  geom_smooth() +
  labs(x = "mpg", y = "Маса автомобіля, 1000 lb",
       title = "Зв’язок mpg та wt")
```

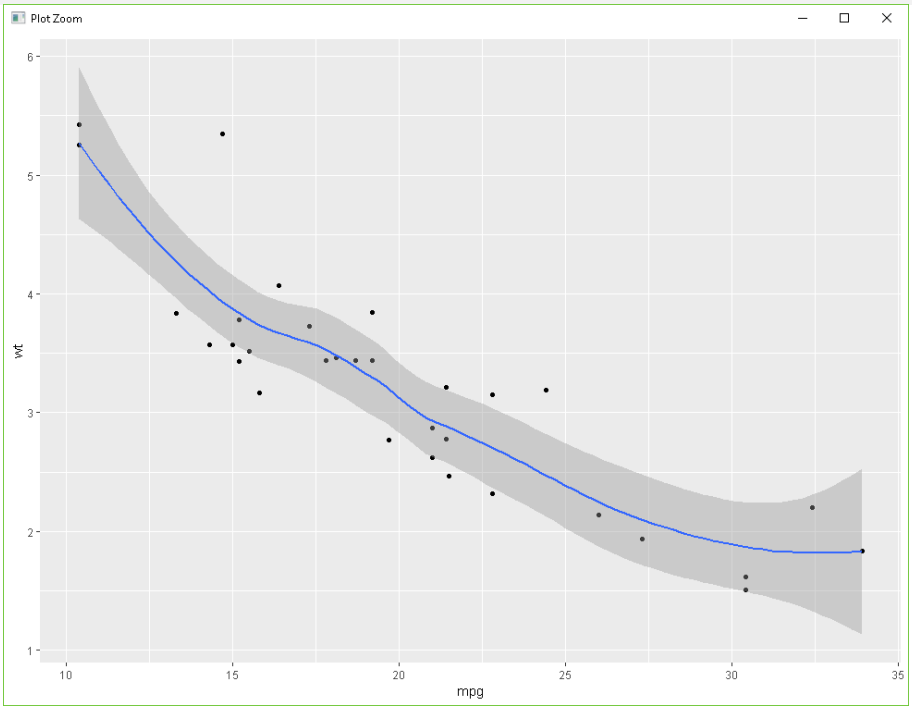


Рис. 5.2. Приклад точкової діаграми зі згладжувальною кривою

3.4. Стовпчасті діаграми

Для дискретної змінної cyl geom_bar() будує стовпчасту діаграму частот. Це не гістограма в статистичному розумінні.

```
ggplot(mtcars, aes(x = factor(cyl))) +
  geom_bar() +
  labs(x = "Кількість циліндрів", y = "Кількість автомобілів")
```

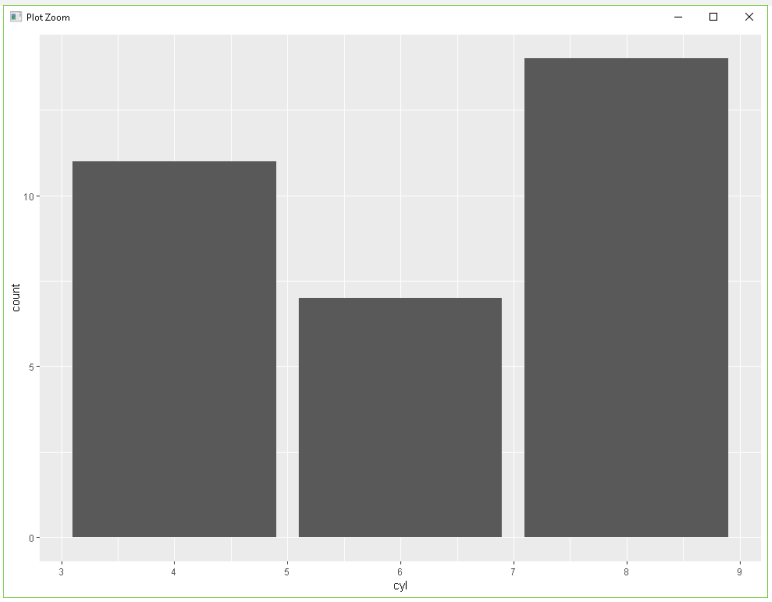


Рис. 5.3. Частоти автомобілів за кількістю циліндрів

3.5. Відображення кількох категорій

```
ggplot(diamonds, aes(x = clarity, fill = cut)) +
  geom_bar() +
  labs(x = "Чистота", y = "Кількість діамантів", fill = "Огранювання",
       title = "Розподіл діамантів за чистотою та якістю огранювання")
+
  theme_bw()
```

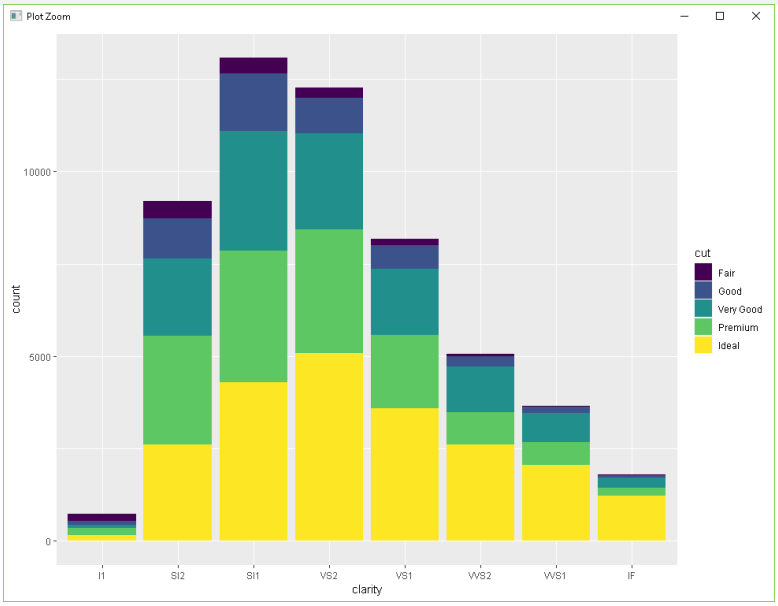


Рис. 5.4. Стовпчаста діаграма для набору diamonds

3.6. Координати й оформлення

coord_polar() можна використати для демонстрації зміни координатної системи, але полярне подання не є автоматично кращим за декартове. Тип візуалізації обирають відповідно до аналітичної задачі та читабельності.

```
ggplot(mtcars, aes(x = factor(cyl))) +
  geom_bar() +
  coord_polar()

ggplot(mtcars, aes(x = mpg, y = disp)) +
  geom_point() +
  theme_bw()
```

Завдання до частини 3

1. Підключити ggplot2 та перевірити структуру наборів mtcars і diamonds.

2. Побудувати діаграму розсіювання для двох кількісних змінних; додати змістовні назви осей і заголовок.

3. Додати згладжувальну криву та пояснити, яку додаткову інформацію вона дає.

4. Побудувати стовпчасту діаграму для дискретної змінної та правильно назвати тип графіка.

5. Побудувати графік для diamonds з використанням щонайменше двох естетичних відображень, наприклад x і fill.

6. Змінити тему графіка та аргументовано пояснити, чи покращилася читабельність.

7. За власними даними побудувати щонайменше три різні змістовні візуалізації та коротко інтерпретувати кожну.

Методичні вказівки до виконання самостійної роботи:

Вирішення прикладних кейсових завдань, висунутих на самостійне опрацювання, повністю відповідає за алгоритмом виконання, технологічним стеком коду, послідовністю команд мови R в інфраструктурі RStudio, а також логіці інтерпретації результатів

тим крокам, які були детально вивчені та продемонстровані під час опрацювання базового навчального масиву безпосередньо на лабораторній роботі.

Індивідуальне дослідження вимагає відтворення наскрізної

архітектури аналізу даних: від написання відтворюваного коду імпорту та первинного очищення матриці до статистичного тестування розподілів (shapiro.test), побудови лінійної моделі (lm) та генерації багатошарових аналітичних графіків екосистеми ggplot2.

ВАРІАНТИ КЕЙСОВИХ ЗАВДАНЬ ДЛЯ САМОСТІЙНОГО ВИРІШЕННЯ:

Під час написання коду здобувач має інтегрувати у функцію прогнозування індивідуальні значення предикторів, наведені в таблиці:

Параметр конфігурації	Кейс 1 (FinTech)	Кейс 2 (PropTech)	Кейс 3 (Marketing)	Кейс 4 (Логістика)	Кейс 5 (Green Economy)
Обсяг вибірки (N)	25	30	25	30	25
Метод кореляції	Pearson	Pearson	Pearson	Pearson	Pearson
Метод тренду в ggplot2	"lm"	"lm"	"loess"	"loess"	"lm"
Базова тема графіка	theme_bw()	theme_minimal()	theme_classic()	theme_light()	theme_linedraw()
Нове значення X для прогнозу №1	15.0	45.0	50.0	3.5	40.0
Нове значення X для прогнозу №2	85.0	120.0	250.0	18.5	150.0
Рівень довіри інтервалу (Level)	0.95	0.95	0.95	0.95	0.95

Для успішного розв'язання самостійного завдання здобувач вищої освіти має послідовно виконати такі етапи:

1. Конфігурація середовища розробки: Створити чистий файл скрипта .R або блокнот .Rmd/.qmd в RStudio. Задати чітку структуру коментарів за допомогою символу #.

2. Імпорт та підготовка даних (Частина 1): Сформувати вхідний датафрейм data.frame відповідно до параметрів свого варіанта (обсяг вибірки N = 25 або N = 30 спостережень). Перевірити внутрішню структуру масиву за допомогою команд str(), dim() та summary().

3. Експлораторний аналіз та тестування (Частина 2): Розрахувати середні значення та показники середньоквадратичного відхилення (sd) для обох числових змінних. Провести графічну (Q-Q plot) та математичну перевірку форми розподілу вибірки на відповідність критерію нормальності Шапіро-Вілка (shapiro.test).

4. Економетричне моделювання: Розрахувати коефіцієнт парної кореляції Пірсона (cor.test) та побудувати лінійне рівняння регресії за допомогою функції lm(). Провести критичний аудит параметрів R^2, t-статистик та рівнів значущості p-value.

5. Діагностика та інтервальний прогноз: Викликати 4 стандартні графіки діагностики залишків за допомогою команди plot(model). Розрахувати точковий та інтервальний прогноз (predict(..., interval = "prediction")) для двох нових значень предиктора, висунутих у завданні.

6. Багатовимірна візуалізація (Частина 3): Засобами пакету ggplot2 згенерувати щонайменше 3 різні змістовні візуалізації (діаграму розсіювання з лінією тренду geom_smooth(), стовпчасту діаграму частот geom_bar()). Обов'язково налаштувати естетичні відображення кольору/заповнення (color, fill), додати підписи осей, заголовки та оптимізувати читабельність за допомогою тем (theme_bw() або theme_minimal()).

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Кучеров Д.П. Методи аналізу великих даних «Big Data» / Д.П. Кучеров. – Київ. – 2020 - 237 с.
2. Ланде Д. В. Оброблення надвеликих масивів даних (Big Data) [Електронний ресурс] : навчальний посібник для використання у навчальному процесі з підготовки фахівців другого (магістерського) рівня вищої освіти зі спеціальності 122 «Комп'ютерні науки» / Д. В. Ланде, І. Ю. Субач, А. Я. Гладун ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 168 с
3. Навчальний посібник з дисципліни "Технології Big Data" для студентів спеціальності 123 - "Комп'ютерна інженерія" / Таран В.І., Гордієнко Ю.Г., Стіренко С.Г. - Київ: КПІ, 2022. - 56 с.
4. Технології оброблення великих даних: конспект лекцій з дисципліни «Технології оброблення великих даних» [Електронний ресурс] : навч. посіб. / Л.М. Олещенко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 227 с.
5. Солодовник, Г. В. Розробка та аналіз алгоритмів [Електронний ресурс]: навчальний посібник / Г. В. Солодовник, О. О. Шаповалова; Харківський національний економічний університет ім. С. Кузнеця. — Електрон. текстові дан. (4,35 МБ). — Харків : ХНЕУ ім. С. Кузнеця, 2023. — 250 с. — Загол. з титул. екрану. — Бібліогр.: с. 223-224. https://library-service.com.ua:8443/khkhneu/DocumentDescription?docid=KhKHNEU.3058872
6. Data Science for Economics and Finance Methodologies and Applications / Ed.: Sergio Consoli, Diego Reforgiato Recupero, Michaela Saisana. Open Access, 2021 https://link.springer.com/book/10.1007/978-3-030-66891-4
7. Security Infrastructure Technology for Integrated Utilization of Big Data. Applied to the Living Safety and Medical Fields / Ed.: Atsuko Miyaji, Tomoaki Mimoto. Open Access, 2020 https://link.springer.com/book/10.1007/978-981-15-3654-0
8. Data Journeys in the Sciences / Ed.: Sabina Leonelli, Niccolò Tempini . Open Access, 2020 https://link.springer.com/book/10.1007/978-3-030-37177-7

Додаткова

9. Базилевич О. В. / Застосування аналізу великих даних для оцінки ефективності рекламної кампанії підприємства / О. В. Базилевич, Ю. Е. Парфьонов // Інформаційні технології та системи : матеріали міжнар. наук.-практ. конф., 8 - 9 квіт. 2021 р.: тези допов. – Харків : ХНЕУ ім. Семена Кузнеця, 2021. – С. 26. URL: http://repository.hneu.edu.ua/handle/123456789/25305

10. Татаринцева Ю. Л. Інноваційний менеджмент в епоху цифрового маркетингу вражень / Ю. Л. Татаринцева, О. І. Пушкар // Актуальні проблеми менеджменту: теоретичні і практичні аспекти : матеріали 6-ї міжнар. наук.–практ. конф., 28-29 вересня 2023 р. : тези допов. – Одеса : ОНЕУ, 2023. – С. 146-148. http://repository.hneu.edu.ua/handle/123456789/30481

11. Матвієнко-Біляєва Г. Л. Тенденції розвитку цифрової економіки як господарської системи / Г. Л. Матвієнко-Біляєва // Економічний простір. –2022. – № 181. – С. 115 – 119. http://repository.hneu.edu.ua/handle/123456789/28767

12. Human-Centered Services Computing for Smart Cities. IEICE Monograph / Ed.: Yohei Murakami, Kosaku Kimura. Open Access, 2024 https://link.springer.com/book/10.1007/978-981-97-0779-9

13. Human-Centered Services Computing for Smart Cities. IEICE Monograph / Ed.: Yohei Murakami, Kosaku Kimura. Open Access, 2024 https://link.springer.com/book/10.1007/978-981-97-0779-9

14. Minukhin S. Performance study of the DTU model for relational databases on the Azure platform / S. Minukhin // Сучасний стан наукових досліджень та технологій в промисловості. – 2022. – № 1 (19). – С. 27-39. URL: http://repository.hneu.edu.ua/handle/123456789/27739

Інформаційні ресурси в Інтернеті

15. Сторінка курсу на платформі Moodle (персональна навчальна система) . – Режим доступу : https://pns.hneu.edu.ua/course/view.php?id=8225

16. Big Data, Hadoop, and Spark Basics. EDX – URL : https://learning.edx.org/course/course-v1:IBM+BD0225EN+2T2021/home

17. A brief introduction to two data processing architectures — Lambda and Kappa for Big Data // Електронний ресурс. Режим доступу: https://medium.com/data-science/a-brief-introduction-to-two-data-processing-architectures-lambda-and-kappa-for-big-data-4f35c28005bb

18. IoT Fundamentals: Big Data & Analytics // Електронний ресурс. Режим доступу: https://www.netacad.com/courses/iot/big-data-analytics

НАВЧАЛЬНЕ ВИДАННЯ

МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗУ BIG DATA:

методичні рекомендації до виконання лабораторних та самостійних робіт

для здобувачів вищої освіти спеціальності - С1 "Економіка та міжнародні відносини" (за спеціалізацією С1.01.Економіка), освітньої програми "Аналітика даних в економіці" другого (магістерського) рівня

Самостійне електронне текстове мережеве видання

Укладачі: **Раєвнєва** Олена Валентинівна
Стрижиченко Костянтин Анатолійович
Бровко Ольга Іванівна

Відповідальний за випуск *О. В. Раєвнєва*

Редактор

Коректор

План 2025 р. Поз. № 24 ЕВ Обсяг 63 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру ***ДК № 4853 від 20.02.2015 р.***