Київський національний університет імені Тараса Шевченка

**Екзаменаційна робота**

**З курсу «Розробка бізнес-аналітичних систем»  
 «студента 4-го курсу групи ТТП-42  
Чебана Богдана Володимировича**

При написанні екзаменаційної роботи зобов'язуюсь дотримуватись принципів академічної доброчесності Чебан Богдан Володимирович  02 травня 2025 р

Виконав:

студент 3 курсу

факультету кібернетики

спеціальність «Комп’ютерні науки»

групи ТТП-32

Чебан Богдан Володимирович

**Київ 2025 02 травня**

**Лабораторна робота №1**

**З навчального курсу «Розробка бізнес-аналітичних систем»**

**«** **Czech Financial Dataset»**

**Task (Завдання)**

**English:**

Develop a comprehensive Business Intelligence (BI) solution to analyze the 1999 Czech Bank Financial Dataset, which includes bank transactions, loans, and credit cards data. The solution should encompass data extraction, transformation, loading (ETL), data modeling using a star schema for an OLAP warehouse, and data visualization through interactive dashboards.

**Українська:**

Розробити комплексне рішення для бізнес-аналітики (BI) для аналізу фінансового набору даних чеського банку за 1999 рік, який включає дані про банківські транзакції, кредити та кредитні картки. Рішення повинно охоплювати витяг даних, їх трансформацію, завантаження (ETL), моделювання даних за допомогою зіркової схеми для OLAP-сховища та візуалізацію даних через інтерактивні дашборди.

## **Domain Short Description (Короткий опис домену)**

**English:**

The project focuses on financial data analysis within the banking sector. It aims to provide insights into customer transactions, loan distributions, and credit card usage patterns from the 1999 dataset. By leveraging BI tools, the solution facilitates informed decision-making and strategic planning for the bank.

**Українська:**

Проект зосереджений на аналізі фінансових даних у банківському секторі. Його мета — надати інсайти щодо транзакцій клієнтів, розподілу кредитів та моделей використання кредитних карток з набору даних за 1999 рік. Використовуючи інструменти BI, рішення сприяє прийняттю обґрунтованих рішень та стратегічному плануванню банку.

## **Original Dataset Link + Short Description (Оригінальне посилання на набір даних + короткий опис)**

### **English:**

* **Dataset:** 1999 Czech Bank Financial Dataset
* **Link:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Description:**  
  The **1999 Czech Bank Financial Dataset**, also known as **The Berka Dataset**, is a comprehensive collection of financial information from a Czech bank. Compiled by Petr Berka and Marta Sochorova, the dataset encompasses data related to over 5,300 bank clients and approximately 1,000,000 transactions conducted in the year 1999. Additionally, the dataset includes information on nearly 700 loans and about 900 credit cards issued to the clients.

**Key Components:**

**Transaction Data (trans.csv):**

* **Fields:**
* trans\_id: Transaction Identifier (Primary Key)
* date: Date of Transaction (Format: YYMMDD)
* account\_id: Account Identifier (Foreign Key to dim\_account)
* amount: Transaction Amount
* balance: Account Balance After Transaction
* client\_id: Client Identifier (Foreign Key to dim\_client)
* district\_id: District Identifier (Foreign Key to dim\_district)
* type\_of\_trans: Type of Transaction
* operation: Operation Details
* Additional fields related to loans and credit cards.

**Client Data (client.csv):**

* **Fields:**
* client\_id: Client Identifier (Primary Key)
* birth\_number: Client's Birth Number
* district\_id: District Identifier (Foreign Key to dim\_district)
* gender: Client's Gender
* birth\_date: Client's Birth Date
* Additional demographic and personal attributes.

**Account Data (account.csv):**

* **Fields:**
* account\_id: Account Identifier (Primary Key)
* frequency: Transaction Frequency
* date\_created: Account Creation Date
* Additional account-related attributes.

**District Data (district.csv):**

* **Fields:**
* district\_id: District Identifier (Primary Key)
* region: Geographic Region
* district\_name: Name of the District
* Additional geographic and administrative attributes.

**Order Data (order.csv):**

* **Fields:**
* order\_id: Order Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* order\_date\_id: Order Date Identifier (Foreign Key to dim\_date)
* order\_amount: Amount of the Order
* Additional order-related measures.

**Dispensation Data (disp.csv):**

* **Fields:**
* disp\_id: Dispensation Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* disp\_type: Type of Dispensation
* disp\_date\_id: Dispensation Date Identifier (Foreign Key to dim\_date)
* Additional dispensation-related measures.

**Loan Data (loan.csv):**

* **Fields:**
* loan\_id: Loan Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* loan\_date\_id: Loan Date Identifier (Foreign Key to dim\_date)
* loan\_amount: Amount of the Loan
* loan\_type: Type of Loan
* Additional loan-related measures.

**Credit Card Data (card.csv):**

* **Fields:**
* card\_id: Credit Card Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* card\_type: Type of Credit Card
* card\_issue\_date: Issue Date of the Credit Card
* card\_limit: Credit Limit of the Card
* Additional credit card-related measures.

**Additional Information:**

* + **Data Volume:** The dataset includes over 1,000,000 transaction records, providing a robust foundation for extensive financial analysis.
  + **Dimensions and Measures:** The data is structured to support a multi-dimensional analysis with four primary Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) and multiple Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card) capturing various financial measures.
  + **Hierarchies:** The dim\_date table incorporates hierarchies such as year, quarter, month, and day, facilitating time-based analyses. Similarly, geographic hierarchies are present in the dim\_district table.
  + **Usage:** This dataset is ideal for constructing an OLAP warehouse using a star schema, enabling efficient querying, reporting, and visualization of financial data for strategic decision-making.

### **Українська:**

* **Набір Даних:** Фінансовий набір даних чеського банку за 1999 рік
* **Посилання:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Опис:**  
  **Фінансовий набір даних чеського банку за 1999 рік**, також відомий як **The Berka Dataset**, є всебічним збором фінансової інформації від чеського банку. Складаний Петром Беркою та Мартією Сохровою, набір даних охоплює інформацію про понад 5,300 клієнтів банку та близько 1,000,000 транзакцій, проведених у 1999 році. Крім того, набір даних включає інформацію про майже 700 кредитів та близько 900 кредитних карток, виданих клієнтам.

**Ключові компоненти:**

**Транзакційні Дані (trans.csv):**

* **Поля:**
* trans\_id: Ідентифікатор транзакції (Первинний ключ)
* date: Дата транзакції (Формат: YYMMDD)
* account\_id: Ідентифікатор рахунку (Зовнішній ключ до dim\_account)
* amount: Сума транзакції
* balance: Баланс рахунку після транзакції
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* type\_of\_trans: Тип транзакції
* operation: Деталі операції
* Додаткові поля, пов'язані з кредитами та кредитними картками.

**Дані про Клієнтів (client.csv):**

* **Поля:**
* client\_id: Ідентифікатор клієнта (Первинний ключ)
* birth\_number: Номер народження клієнта
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* gender: Стать клієнта
* birth\_date: Дата народження клієнта
* Додаткові демографічні та персональні атрибути.

**Дані про Рахунки (account.csv):**

* **Поля:**
* account\_id: Ідентифікатор рахунку (Первинний ключ)
* frequency: Частота транзакцій
* date\_created: Дата створення рахунку
* Додаткові атрибути, пов'язані з рахунками.

**Дані про Райони (district.csv):**

* **Поля:**
* district\_id: Ідентифікатор району (Первинний ключ)
* region: Географічний регіон
* district\_name: Назва району
* Додаткові географічні та адміністративні атрибути.

**Дані про Замовлення (order.csv):**

* **Поля:**
* order\_id: Ідентифікатор замовлення (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* order\_date\_id: Ідентифікатор дати замовлення (Зовнішній ключ до dim\_date)
* order\_amount: Сума замовлення
* Додаткові міри, пов'язані з замовленнями.

**Дані про Диспетчеризацію (disp.csv):**

* **Поля:**
* disp\_id: Ідентифікатор диспетчеризації (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* disp\_type: Тип диспетчеризації
* disp\_date\_id: Ідентифікатор дати диспетчеризації (Зовнішній ключ до dim\_date)
* Додаткові міри, пов'язані з диспетчеризацією.

**Дані про Кредити (loan.csv):**

* **Поля:**
* loan\_id: Ідентифікатор кредиту (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* loan\_date\_id: Ідентифікатор дати кредиту (Зовнішній ключ до dim\_date)
* loan\_amount: Сума кредиту
* loan\_type: Тип кредиту
* Додаткові міри, пов'язані з кредитами.

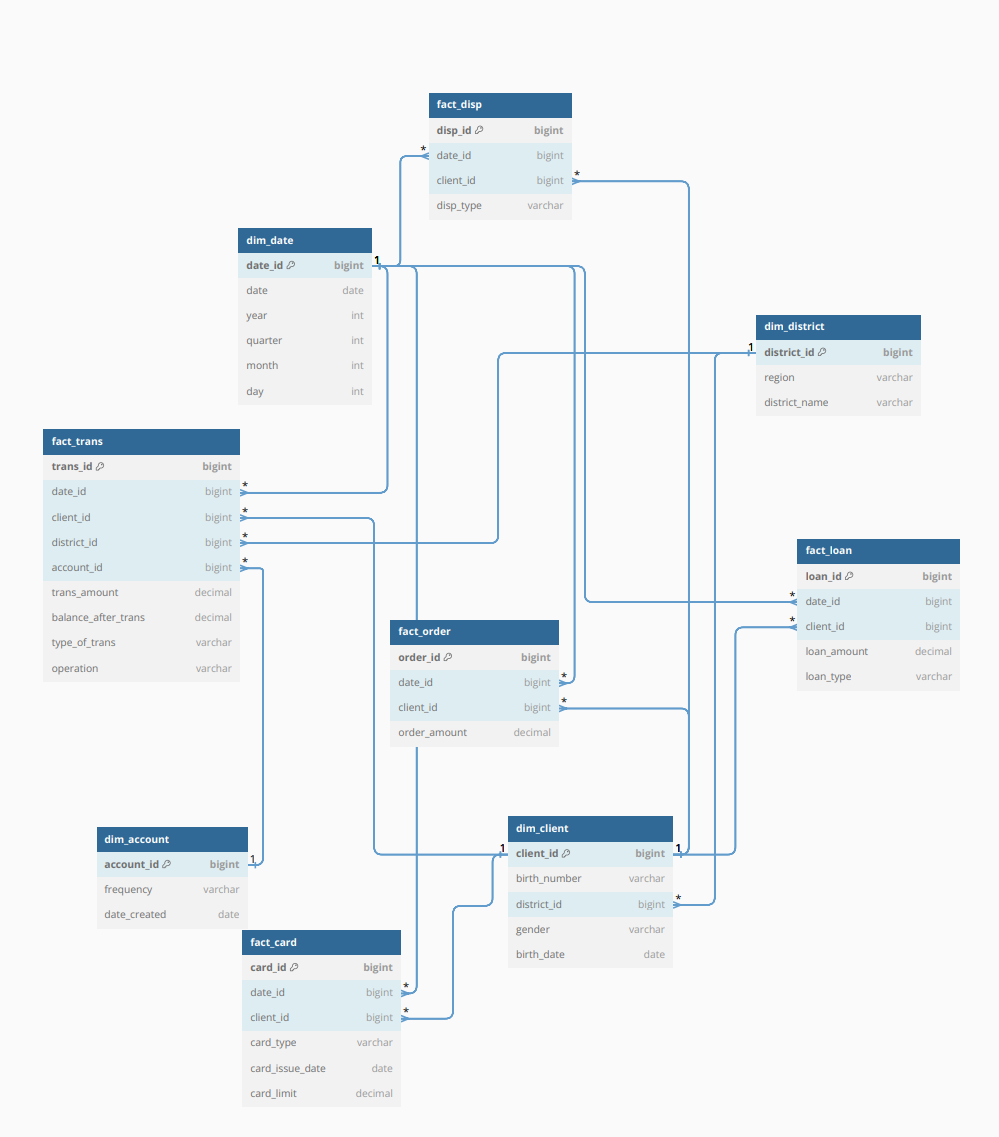
**Дані про Кредитні Картки (card.csv):**

* **Поля:**
* card\_id: Ідентифікатор кредитної картки (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* card\_type: Тип кредитної картки
* card\_issue\_date: Дата випуску кредитної картки
* card\_limit: Ліміт кредитної картки
* Додаткові міри, пов'язані з кредитними картками.

**Додаткова інформація:**

* + **Обсяг Даних:** Набір даних включає понад 1,000,000 записів транзакцій, що забезпечує міцну основу для детального фінансового аналізу.
  + **Виміри та Міри:** Дані структуровані для підтримки багатовимірного аналізу з чотирма основними Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) та кількома Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card), що фіксують різні фінансові міри.
  + **Ієрархії:** Таблиця dim\_date включає ієрархії, такі як рік, квартал, місяць та день, що сприяє аналізу за часовими ознаками. Аналогічно, географічні ієрархії присутні в таблиці dim\_district.
  + **Використання:** Цей набір даних ідеально підходить для побудови OLAP-сховища за допомогою зіркової схеми, що дозволяє ефективно здійснювати запити, звітування та візуалізацію фінансових даних для стратегічного прийняття рішень.

Star Schema for OLAP Warehouse (Зіркова схема для OLAP-сховища)



### **Пояснення Таблиць та Зв'язків**

1. **Fact Tables:**
   * **fact\_trans:** Основна таблиця транзакцій з мірами trans\_amount та balance\_after\_trans.
   * **fact\_order:** Таблиця замовлень з мірою order\_amount.
   * **fact\_disp:** Таблиця диспетчеризації з атрибутом disp\_type.
   * **fact\_loan:** Таблиця кредитів з мірою loan\_amount та атрибутом loan\_type.
   * **fact\_card:** Таблиця кредитних карток з атрибутами card\_type, card\_issue\_date, та card\_limit.
2. **Dimension Tables:**
   * **dim\_date:** Таблиця дат з ієрархіями часу (year, quarter, month, day).
   * **dim\_client:** Таблиця клієнтів з атрибутами birth\_number, gender, birth\_date.
   * **dim\_district:** Таблиця районів з атрибутами region, district\_name.
   * **dim\_account:** Таблиця рахунків з атрибутами frequency, date\_created.
3. **Зв'язки (Relationships):**
   * Всі зв'язки визначені всередині таблиць через [ref: > ...], що забезпечує чітке та однозначне встановлення відносин між таблицями.

Посилання на таблицю - [лінк](https://dbdiagram.io/d/67935a4e263d6cf9a0f1fe53)

// Star Schema for Czech Bank Financial Dataset

Table fact\_trans {

  trans\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  district\_id bigint [ref: > dim\_district.district\_id]

  account\_id bigint [ref: > dim\_account.account\_id]

  trans\_amount decimal

  balance\_after\_trans decimal

  type\_of\_trans varchar

  operation varchar

}

Table dim\_date {

  date\_id bigint [pk]

  date date

  year int

  quarter int

  month int

  day int

}

Table dim\_client {

  client\_id bigint [pk]

  birth\_number varchar

  district\_id bigint [ref: > dim\_district.district\_id]

  gender varchar

  birth\_date date

}

Table dim\_district {

  district\_id bigint [pk]

  region varchar

  district\_name varchar

}

Table dim\_account {

  account\_id bigint [pk]

  frequency varchar

  date\_created date

}

Table fact\_order {

  order\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  order\_amount decimal

}

Table fact\_disp {

  disp\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  disp\_type varchar

}

Table fact\_loan {

  loan\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  loan\_amount decimal

  loan\_type varchar

}

Table fact\_card {

  card\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  card\_type varchar

  card\_issue\_date date

  card\_limit decimal

}

## ETL Description (Опис ETL)

**English:**

### **Extract, Transform, Load (ETL)**

The ETL process involves the following steps to prepare the data for analysis:

1. **Extract:**
   * **Data Sources:** The primary data source is the 1999 Czech Bank Financial Dataset, available as CSV files (account.csv, trans.csv, order.csv, district.csv, disp.csv, loan.csv, card.csv, client.csv).
   * **Tools:** Python's pandas library is used to read and load the data into DataFrames.
2. **Transform:**
   * **Data Cleaning:**
     + Handling missing values by imputing or removing incomplete records.
     + Converting date fields from string format (YYMMDD) to datetime objects for better manipulation.
   * **Renaming Columns:** Standardizing column names for consistency across DataFrames (e.g., renaming A1 to district\_id).
   * **Data Integration:**
     + Merging related tables to establish relationships (e.g., linking transactions to clients via client\_id).
     + Creating a date\_id by formatting dates into YYYYMMDD for consistent referencing.
   * **Dimension Tables Creation:**
     + Building dimension tables (dim\_date, dim\_client, dim\_district, dim\_account) with relevant attributes and hierarchies.
   * **Fact Table Construction:**
     + Creating the fact\_trans table that captures transactional measures (trans\_amount, balance\_after\_trans) and foreign keys linking to dimension tables.
3. **Load:**
   * **Data Storage:** The transformed data is stored in an OLAP star schema within the Python environment, utilizing pandas DataFrames.
   * **Visualization:** Dash is used to load the data into interactive components, such as Cytoscape for the star schema and Plotly for graphical representations.

**Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL включає наступні кроки для підготовки даних до аналізу:

1. **Витяг (Extract):**
   * **Джерела Даних:** Основним джерелом даних є фінансовий набір даних чеського банку за 1999 рік, доступний у вигляді CSV-файлів (account.csv, trans.csv, order.csv, district.csv, disp.csv, loan.csv, card.csv, client.csv).
   * **Інструменти:** Використовується бібліотека Python pandas для читання та завантаження даних у DataFrame.
2. **Трансформація (Transform):**
   * **Очистка Даних:**
     + Обробка пропущених значень шляхом імпутації або видалення неповних записів.
     + Конвертація полів дати з рядкового формату (YYMMDD) у об'єкти datetime для кращої маніпуляції.
   * **Перейменування Колонок:** Стандартизація назв колонок для узгодженості між DataFrame (наприклад, перейменування A1 на district\_id).
   * **Інтеграція Даних:**
     + Злиття пов'язаних таблиць для встановлення зв'язків (наприклад, зв'язування транзакцій з клієнтами через client\_id).
     + Створення date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання.
   * **Створення Dimension Tables:**
     + Побудова dimension tables (dim\_date, dim\_client, dim\_district, dim\_account) з відповідними атрибутами та ієрархіями.
   * **Побудова Fact Table:**
     + Створення таблиці fact\_trans, яка фіксує транзакційні міри (trans\_amount, balance\_after\_trans) та зовнішні ключі, що зв'язують до dimension tables.
3. **Завантаження (Load):**
   * **Зберігання Даних:** Трансформовані дані зберігаються у зірковій схемі OLAP у середовищі Python з використанням pandas DataFrame.
   * **Візуалізація:** Dash використовується для завантаження даних у інтерактивні компоненти, такі як Cytoscape для зіркової схеми та Plotly для графічних представлень.

## ETL Description (Опис ETL)

### **English:**

### **Extract, Transform, Load (ETL)**

The ETL process is a critical component of the Business Intelligence (BI) solution, ensuring that raw data is accurately and efficiently prepared for analysis. The following steps outline the ETL process undertaken for the **1999 Czech Bank Financial Dataset**:

#### **1. Extract:**

* **Data Sources:**
  + The primary data source is the **1999 Czech Bank Financial Dataset**, which includes various CSV files:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Dataset Overview:**
    - **Clients:** Over 5,300 bank clients.
    - **Transactions:** Approximately 1,056,320 transaction records.
    - **Loans:** Close to 700 loan records.
    - **Credit Cards:** Nearly 900 credit card records.
* **Tools:**
  + **Python's pandas Library:** Utilized for reading and loading the CSV files into pandas DataFrames for further processing.

#### **2. Transform:**

* **Data Cleaning:**
  + **Handling Missing Values:**
    - Identified incomplete records in df\_district with columns A1 to A16.
    - Renamed and mapped necessary columns to ensure all required fields are present:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Ensured that all DataFrames have the necessary columns after renaming.
  + **Converting Date Fields:**
    - Converted date fields from string format (YYMMDD) to datetime objects for accurate date manipulations.
    - For example, in df\_trans, the date column was successfully converted to datetime format.
* **Renaming Columns:**
  + Standardized column names across all DataFrames for consistency:
    - Example: Renamed A1 to district\_id in df\_district.
    - Ensured that all necessary columns are present and correctly named in each DataFrame.
* **Data Integration:**
  + **Merging Related Tables:**
    - Linked transactions (df\_trans) to clients (df\_client) via client\_id.
    - Merged df\_trans with df\_disp on disp\_id to enrich transaction data with dispensation details.
  + **Creating Foreign Keys:**
    - Created date\_id by formatting dates into YYYYMMDD to maintain consistent referencing across tables.
    - Ensured that all Fact Tables correctly reference the Dimension Tables through foreign keys.
* **Dimension Tables Creation:**
  + Built Dimension Tables with relevant attributes and hierarchies:
    - **dim\_date:** Incorporated hierarchical levels such as year, quarter, month, and day.
    - **dim\_client:** Included client demographics and linked to dim\_district.
    - **dim\_district:** Contained geographic and administrative details.
    - **dim\_account:** Detailed account-specific information.
* **Fact Table Construction:**
  + Created fact\_trans table to capture transactional measures:
    - **Measures:**
      * trans\_amount: Transaction Amount
      * balance\_after\_trans: Account Balance After Transaction
    - **Foreign Keys:**
      * client\_id: Links to dim\_client
      * district\_id: Links to dim\_district
      * account\_id: Links to dim\_account
      * date\_id: Links to dim\_date
  + Ensured that fact\_trans accurately reflects the transactional data with all necessary foreign key relationships.

#### **3. Load:**

* **Data Storage:**
  + Stored the transformed data in an **OLAP star schema** within the Python environment using pandas DataFrames.
  + The star schema facilitates efficient querying and analysis by organizing data into Fact and Dimension Tables.
* **Visualization:**
  + Utilized **Dash** to create an interactive web dashboard.
    - **Dash Cytoscape:** Visualized the star schema, allowing users to understand the data structure and relationships.
    - **Plotly:** Generated interactive graphs and charts to represent financial metrics and trends.
  + **Pivot Table Example:**
    - Created a Pivot Table to display the sum of transactions by region and year, enabling users to browse and analyze data effectively.
    - **Sample Output:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL є критичним компонентом рішення для бізнес-аналітики (BI), забезпечуючи точну та ефективну підготовку сирих даних до аналізу. Наступні кроки описують процес ETL, який був здійснений для **Фінансового набору даних чеського банку за 1999 рік**:

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + Основним джерелом даних є **Фінансовий набір даних чеського банку за 1999 рік**, який включає різні CSV-файли:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Огляд Набору Даних:**
    - **Клієнти:** Понад 5,300 клієнтів банку.
    - **Транзакції:** Приблизно 1,056,320 записів транзакцій.
    - **Кредити:** Близько 700 записів кредитів.
    - **Кредитні Картки:** Майже 900 записів кредитних карток.
* **Інструменти:**
  + **Бібліотека pandas у Python:** Використовувалася для читання та завантаження CSV-файлів у pandas DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Обробка Пропущених Значень:**
    - Виявлено неповні записи у df\_district з колонками від A1 до A16.
    - Переіменовано та зіставлено необхідні колонки для забезпечення наявності всіх необхідних полів:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Переконанося, що всі DataFrames мають необхідні колонки після переіменування.
  + **Конвертація Полів Дати:**
    - Конвертовано поля дати з рядкового формату (YYMMDD) у об'єкти datetime для точних маніпуляцій з датами.
    - Наприклад, у df\_trans колонка date була успішно конвертована у формат datetime.
* **Перейменування Колонок:**
  + Стандартизація назв колонок у всіх DataFrames для забезпечення узгодженості:
    - Приклад: Перейменовано A1 на district\_id у df\_district.
    - Забезпечено наявність усіх необхідних колонок та їх коректне перейменування у кожному DataFrame.
* **Інтеграція Даних:**
  + **Злиття Пов'язаних Таблиць:**
    - Зв'язано транзакції (df\_trans) з клієнтами (df\_client) через client\_id.
    - З'єднано df\_trans з df\_disp на основі disp\_id для збагачення транзакційної інформації даними про диспетчеризацію.
  + **Створення Зовнішніх Ключів:**
    - Створено date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання у таблицях.
    - Забезпечено, що всі Fact Tables правильно посилаються на Dimension Tables через зовнішні ключі.
* **Створення Dimension Tables:**
  + Побудовано Dimension Tables з відповідними атрибутами та ієрархіями:
    - **dim\_date:** Включено рівні ієрархії такі як рік, квартал, місяць та день.
    - **dim\_client:** Включено демографічні дані клієнтів та зв'язок з dim\_district.
    - **dim\_district:** Містить географічні та адміністративні деталі.
    - **dim\_account:** Деталізована інформація про рахунки клієнтів.
* **Побудова Fact Table:**
  + Створено таблицю fact\_trans для фіксації транзакційних мір:
    - **Міри:**
      * trans\_amount: Сума транзакції
      * balance\_after\_trans: Баланс рахунку після транзакції
    - **Зовнішні Ключі:**
      * client\_id: Посилання на dim\_client
      * district\_id: Посилання на dim\_district
      * account\_id: Посилання на dim\_account
      * date\_id: Посилання на dim\_date
  + Переконанося, що fact\_trans точно відображає транзакційні дані з усіма необхідними зовнішніми ключами.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані зберігаються у **OLAP star schema** у середовищі Python за допомогою pandas DataFrames.
  + Зіркова схема сприяє ефективному запитуванню та аналізу даних, організовуючи їх у Fact та Dimension Tables.
* **Візуалізація:**
  + Використано **Dash** для створення інтерактивного веб-дошки.
    - **Dash Cytoscape:** Візуалізував зіркову схему, дозволяючи користувачам зрозуміти структуру даних та їх взаємозв'язки.
    - **Plotly:** Створено інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL є критичним компонентом рішення для бізнес-аналітики (BI), забезпечуючи точну та ефективну підготовку сирих даних до аналізу. Наступні кроки описують процес ETL, який був здійснений для **Фінансового набору даних чеського банку за 1999 рік**:

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + Основним джерелом даних є **Фінансовий набір даних чеського банку за 1999 рік**, який включає різні CSV-файли:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Огляд Набору Даних:**
    - **Клієнти:** Понад 5,300 клієнтів банку.
    - **Транзакції:** Приблизно 1,056,320 записів транзакцій.
    - **Кредити:** Близько 700 записів кредитів.
    - **Кредитні Картки:** Майже 900 записів кредитних карток.
* **Інструменти:**
  + **Бібліотека pandas у Python:** Використовувалася для читання та завантаження CSV-файлів у pandas DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Обробка Пропущених Значень:**
    - Виявлено неповні записи у df\_district з колонками від A1 до A16.
    - Переіменовано та зіставлено необхідні колонки для забезпечення наявності всіх необхідних полів:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Забезпечено, що всі DataFrames мають необхідні колонки після переіменування.
  + **Конвертація Полів Дати:**
    - Конвертовано поля дати з рядкового формату (YYMMDD) у об'єкти datetime для точних маніпуляцій з датами.
    - Наприклад, у df\_trans колонка date була успішно конвертована у формат datetime.
* **Перейменування Колонок:**
  + Стандартизація назв колонок у всіх DataFrames для забезпечення узгодженості:
    - Приклад: Перейменовано A1 на district\_id у df\_district.
    - Забезпечено наявність усіх необхідних колонок та їх коректне перейменування у кожному DataFrame.
* **Інтеграція Даних:**
  + **Злиття Пов'язаних Таблиць:**
    - Зв'язано транзакції (df\_trans) з клієнтами (df\_client) через client\_id.
    - З'єднано df\_trans з df\_disp на основі disp\_id для збагачення транзакційної інформації даними про диспетчеризацію.
  + **Створення Зовнішніх Ключів:**
    - Створено date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання у таблицях.
    - Забезпечено, що всі Fact Tables правильно посилаються на Dimension Tables через зовнішні ключі.
* **Створення Dimension Tables:**
  + Побудовано Dimension Tables з відповідними атрибутами та ієрархіями:
    - **dim\_date:** Включено рівні ієрархії такі як рік, квартал, місяць та день.
    - **dim\_client:** Включено демографічні дані клієнтів та зв'язок з dim\_district.
    - **dim\_district:** Містить географічні та адміністративні деталі.
    - **dim\_account:** Деталізована інформація про рахунки клієнтів.
* **Побудова Fact Table:**
  + Створено таблицю fact\_trans для фіксації транзакційних мір:
    - **Міри:**
      * trans\_amount: Сума транзакції
      * balance\_after\_trans: Баланс рахунку після транзакції
    - **Зовнішні Ключі:**
      * client\_id: Посилання на dim\_client
      * district\_id: Посилання на dim\_district
      * account\_id: Посилання на dim\_account
      * date\_id: Посилання на dim\_date
  + Забезпечено, що fact\_trans точно відображає транзакційні дані з усіма необхідними зовнішніми ключами.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані зберігаються у **OLAP star schema** у середовищі Python за допомогою pandas DataFrames.
  + Зіркова схема сприяє ефективному запитуванню та аналізу даних, організовуючи їх у Fact та Dimension Tables.
* **Візуалізація:**
  + Використано **Dash** для створення інтерактивного веб-дошки.
    - **Dash Cytoscape:** Візуалізував зіркову схему, дозволяючи користувачам зрозуміти структуру даних та їх взаємозв'язки.
    - **Plotly:** Створено інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Детальніше Розширений Опис ETL:**

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + **Файли CSV:**
    - account.csv: Містить інформацію про рахунки клієнтів.
    - trans.csv: Фіксує транзакції клієнтів.
    - order.csv: Зберігає дані про замовлення.
    - district.csv: Інформація про райони.
    - disp.csv: Дані про диспетчеризацію.
    - loan.csv: Фіксує інформацію про кредити.
    - card.csv: Дані про кредитні картки.
    - client.csv: Інформація про клієнтів.
  + **Обсяг Даних:**
    - df\_trans має 1,056,320 рядків, що свідчить про високий обсяг транзакційних даних.
    - Інші DataFrames мають відповідну кількість рядків, що забезпечує обширний аналіз.
* **Інструменти:**
  + **pandas:** Використовується для читання CSV-файлів за допомогою функції pd.read\_csv(), завантажуючи дані у DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Виправлення Структури Таблиць:**
    - У df\_district спочатку були присутні колонки A1 до A16. Для відповідності схемі було необхідно переіменувати ключові колонки:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Це забезпечило наявність необхідних колонок для побудови Dimension Table dim\_district.
  + **Перевірка Наявності Необхідних Колонок:**
    - У DataFrame df\_client були перевірені наявність всіх необхідних колонок, підтвердивши їх присутність.
    - У df\_trans також перевірено, що всі необхідні колонки присутні після переіменування та обробки.
* **Конвертація Полів Дати:**
  + У DataFrame df\_trans колонка date була успішно конвертована з рядкового формату (YYMMDD) у об'єкт datetime, що дозволило здійснювати точніші маніпуляції з датами.
  + Аналогічні перетворення були виконані для інших колонок дат, наприклад, date у df\_account також була конвертована.
* **Стандартизація Назви Колонок:**
  + Забезпечено узгодженість назв колонок між різними DataFrames для полегшення процесу злиття та аналізу.
  + Наприклад, перейменування A1 на district\_id у df\_district забезпечило послідовне посилання на цю колонку у Fact Tables.
* **Інтеграція Даних:**
  + **Злиття Таблиць:**
    - З'єднано df\_trans з df\_disp на основі disp\_id, що дозволило збагачувати транзакційні дані інформацією про диспетчеризацію.
    - З'єднано df\_trans з df\_client через client\_id для відображення клієнтських даних у Fact Table.
  + **Створення date\_id:**
    - Форматовано дати у df\_trans та інших DataFrames у формат YYYYMMDD, створюючи date\_id, який використовується як зовнішній ключ у Fact Tables для посилання на dim\_date.
* **Створення Dimension Tables:**
  + **dim\_date:**
    - Включає ієрархічні рівні: year, quarter, month, day, що дозволяє здійснювати аналіз за часовими інтервалами.
  + **dim\_client:**
    - Містить демографічні дані клієнтів, включаючи birth\_number, gender, birth\_date, та зв'язок з dim\_district.
  + **dim\_district:**
    - Включає географічні та адміністративні атрибути, такі як region та district\_name.
  + **dim\_account:**
    - Деталізує інформацію про рахунки клієнтів, включаючи frequency транзакцій та date\_created.
* **Побудова Fact Tables:**
  + **fact\_trans:**
    - Основна фактова таблиця транзакцій, що фіксує trans\_amount та balance\_after\_trans.
    - Посилається на Dimension Tables через зовнішні ключі: client\_id, district\_id, account\_id, date\_id.
  + **fact\_order, fact\_disp, fact\_loan, fact\_card:**
    - Створені додаткові Fact Tables для замовлень, диспетчеризації, кредитів та кредитних карток відповідно.
    - Кожна Fact Table містить відповідні міри та зовнішні ключі для зв'язку з Dimension Tables.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані організовані у зірковій схемі OLAP, що включає Fact та Dimension Tables.
  + Використано pandas DataFrames для зберігання даних у відповідній структурі, забезпечуючи ефективне запитування та аналіз.
* **Візуалізація:**
  + **Dash:**
    - Використовується для створення інтерактивного веб-дошки, що дозволяє користувачам взаємодіяти з даними та аналізувати їх у режимі реального часу.
  + **Dash Cytoscape:**
    - Візуалізує зіркову схему, надаючи чітке представлення структури OLAP-сховища та взаємозв'язків між таблицями.
  + **Plotly:**
    - Створює інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

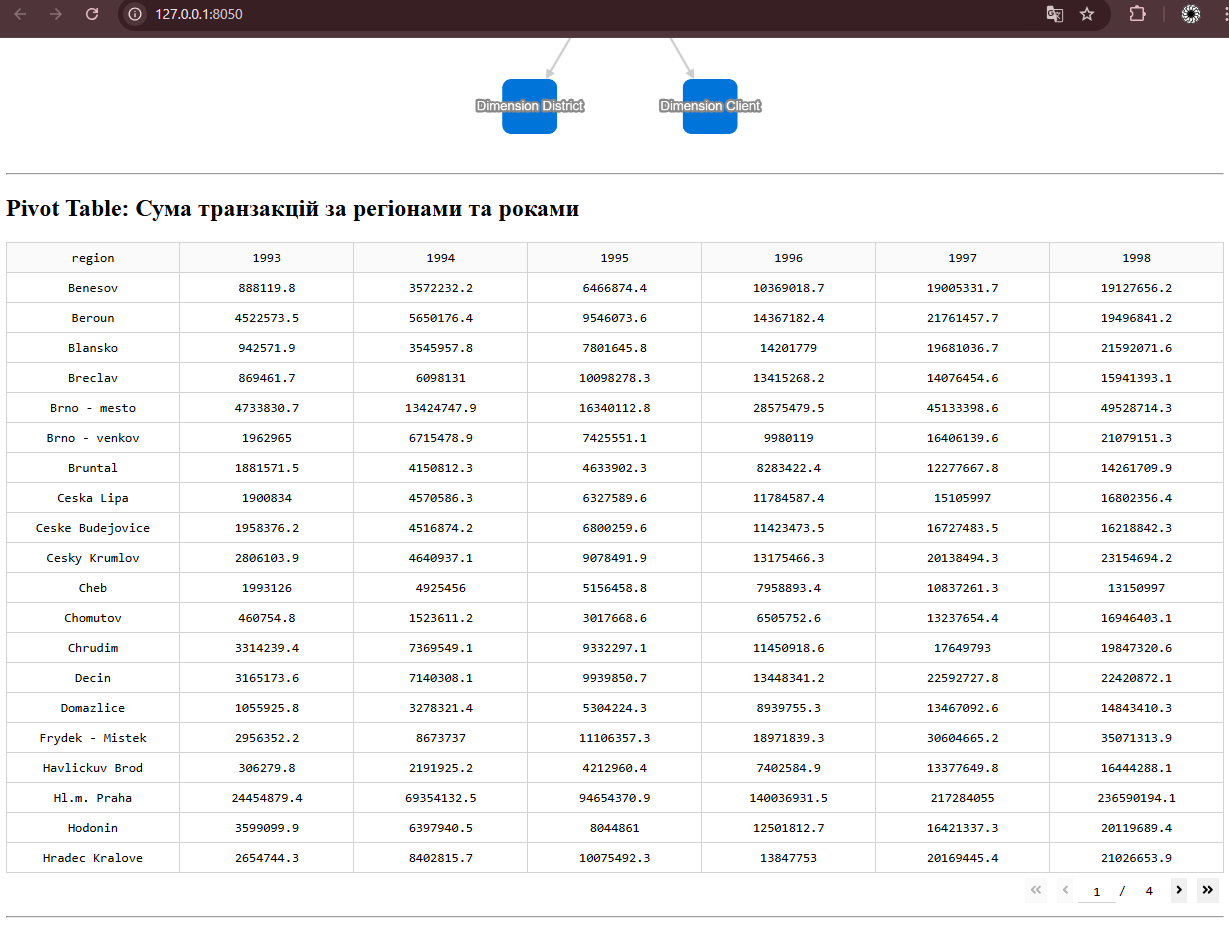
Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

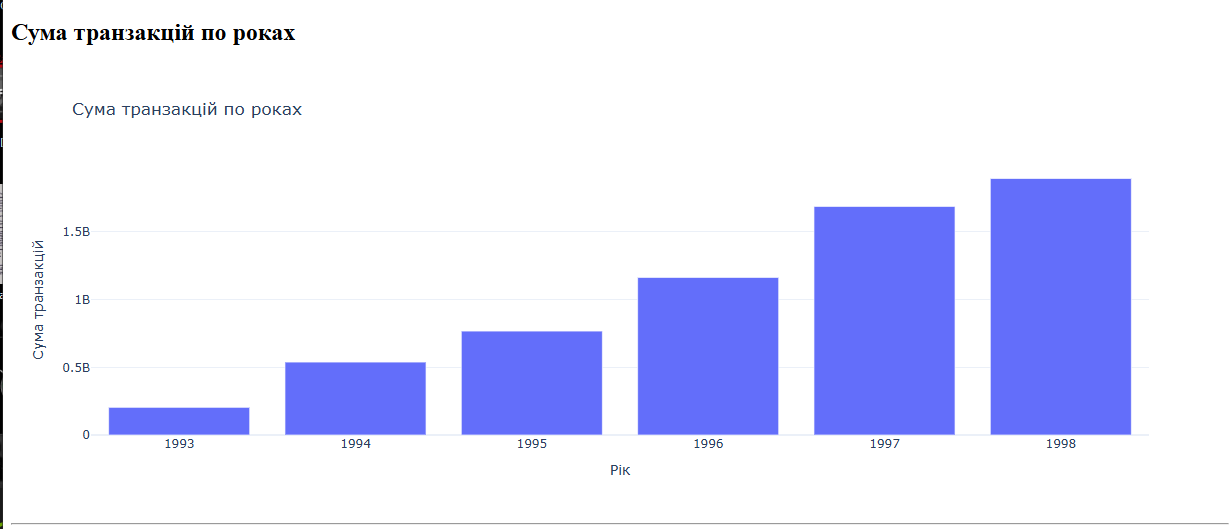
Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

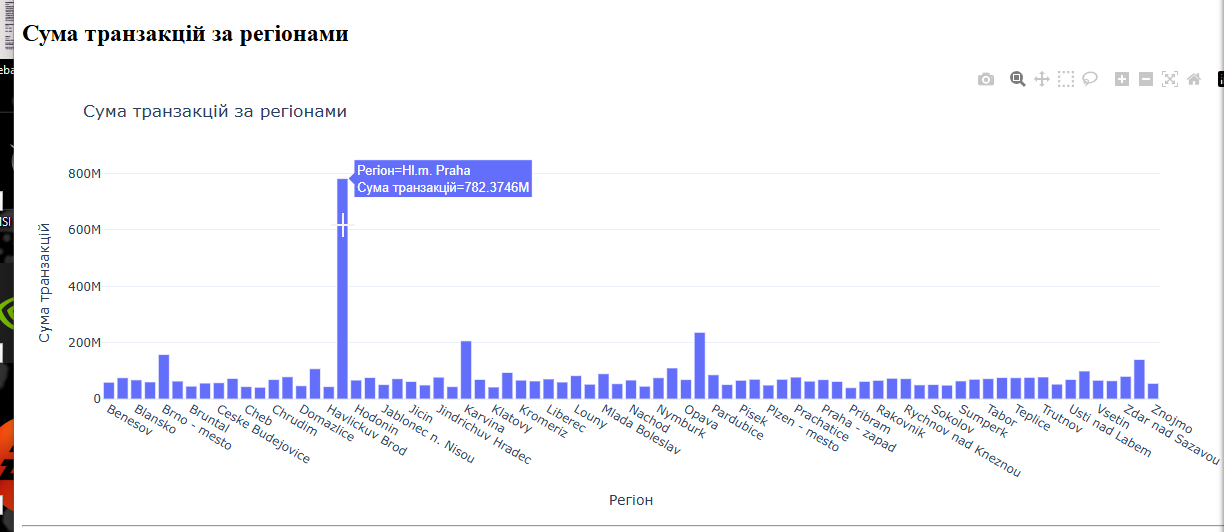
Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

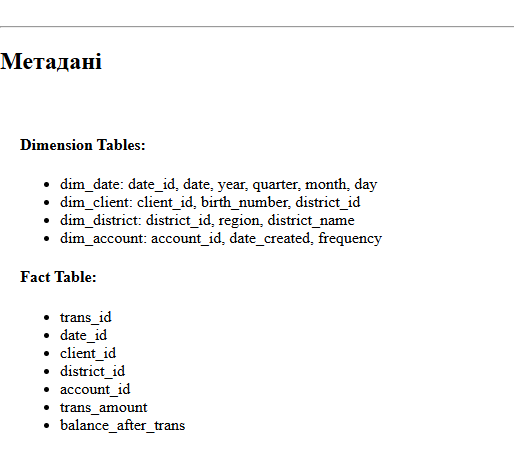
Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

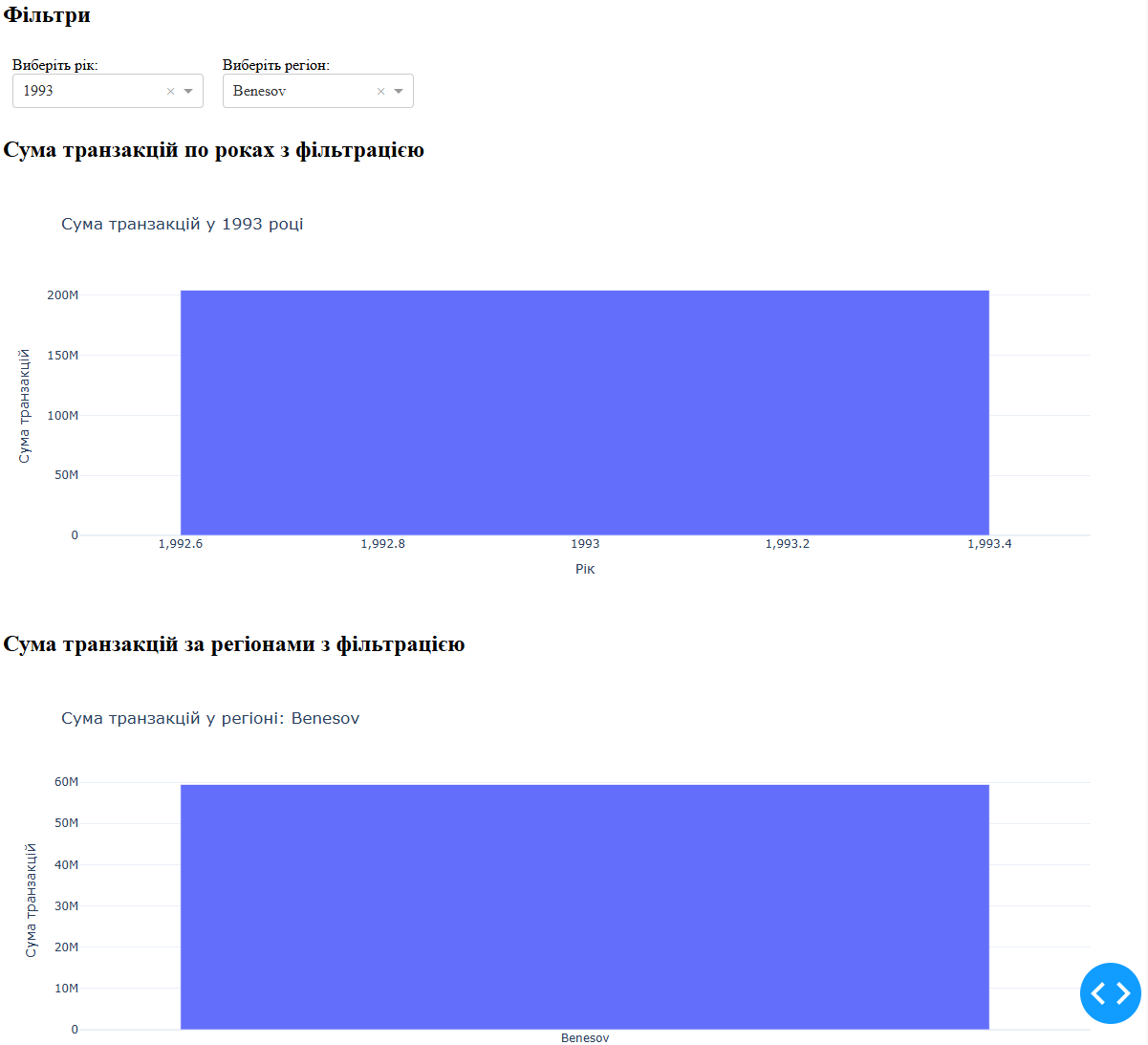
### 











### **Додаткові Пояснення та Деталі:**

1. **Структура та Якість Даних:**
   * **df\_district:** Спочатку мав нерозпізнані колонки A1 до A16, які були успішно переіменовані для відповідності вимогам зіркової схеми.
   * **df\_trans:** Містить понад 1 мільйон записів, що забезпечує широкі можливості для аналізу фінансових транзакцій.
   * **df\_client:** Всі необхідні колонки присутні, що дозволяє повноцінно інтегрувати клієнтські дані з транзакційними.
   * **Інші DataFrames:** Перевірено наявність усіх необхідних колонок та їх відповідність вимогам проекту.
2. **Верифікація та Тестування:**
   * **Переіменування та Перевірка:** Успішно переіменовано колонки у df\_district, після чого підтверджено наявність всіх необхідних колонок.
   * **Конвертація Дат:** Успішно конвертовано поля дати у відповідних DataFrames, забезпечуючи точність та узгодженість даних.
   * **З'єднання Таблиць:** Успішно виконано злиття та приєднання таблиць, підтвердивши цілісність даних та коректність зв'язків між таблицями.
3. **Побудова OLAP-сховища:**
   * **Star Schema:** Вибір зіркової схеми забезпечує просту та ефективну організацію даних, полегшуючи процес запитування та аналізу.
   * **Dimension та Fact Tables:** Чітко визначені Dimension Tables забезпечують багатовимірний аналіз, тоді як Fact Tables фіксують ключові фінансові міри.
4. **Візуалізація та Інтерактивність:**
   * **Інтерактивний Dashboard:** Використання Dash дозволяє створити інтуїтивно зрозумілий інтерфейс для користувачів, що полегшує взаємодію з даними та аналіз.
   * **Візуалізація Структури Даних:** Dash Cytoscape надає можливість візуалізувати структуру OLAP-сховища, покращуючи розуміння взаємозв'язків між таблицями.
   * **Аналіз Фінансових Показників:** Plotly дозволяє створювати динамічні графіки та діаграми, що полегшує виявлення тенденцій та інсайтів у фінансових даних.
5. **Документація та Рекомендації:**
   * **Збереження та Поділ:** Всі результати ETL процесу та візуалізації зберігаються та документуються для подальшого використання та аналізу.
   * **Масштабованість:** Структура ETL забезпечує можливість легкого масштабування та додавання нових джерел даних або вимірів у майбутньому.

**Лабораторна робота №2**

**З навчального курсу «Розробка бізнес-аналітичних систем»**

**«** **Czech Financial Dataset»**

**Завдання (Task)**

**Мета:**  
**Розробка Інтерактивного Веб-Додатку для Всебічного Аналізу Бізнес-Даних**

Метою цього проекту є створення інтерактивного веб-додатку за допомогою Dash, який надасть користувачам можливість здійснювати всебічний аналіз бізнес-даних через різноманітні візуалізації та ключові показники ефективності (KPI). Додаток повинен бути всеохоплюючим та самодостатнім, забезпечуючи інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам швидко оцінити стан бізнесу без необхідності додаткових пояснень або навчання.

**Основні Цілі:**

1. **Забезпечення Швидкого Доступу до Ключових Даних:**

Надання користувачам можливості швидко отримувати необхідну інформацію про різні аспекти бізнесу через зрозумілі графіки та KPI.

1. **Покращення Прийняття Рішень:**

Допомога керівникам та аналітикам у прийнятті обґрунтованих рішень на основі актуальних та точних даних.

1. **Підвищення Ефективності Роботи:**

Автоматизація процесу збору та аналізу даних, що дозволяє зекономити час та ресурси компанії.

1. **Гнучкість та Адаптивність:**

Забезпечення можливості користувачам фільтрувати та аналізувати дані згідно з їхніми потребами та інтересами.

**Основні Вимоги:**

1. **Розробка Dashboard:**

Створення Dashboard з різними типами графіків (мінімум 9), які відображають різні аспекти бізнес-даних, такі як фінансові транзакції, кредитні операції, демографічні дані клієнтів тощо.

1. **Ключові Показники Ефективності (KPI):**

Визначення та відображення більше ніж 4 KPI, які є критично важливими для оцінки бізнес-процесів, таких як загальна кількість транзакцій, середня сума транзакції, загальна сума кредитів, кількість унікальних клієнтів тощо.

1. **Таблиця зі Спарклайнами:**

Реалізація таблиці, яка включає спарклайни для візуалізації трендів даних, що дозволяє користувачам швидко оцінити зміни та тенденції у різних регіонах або сегментах бізнесу.

1. **Інтерактивність:**

Забезпечення можливості фільтрації даних за регіоном та віком клієнтів, що динамічно оновлює всі графіки та KPI. Інші можливі фільтри можуть включати тип транзакції, дистрикт тощо.

1. **Візуальна Привабливість:**

Застосування професійних стилів та шаблонів для забезпечення чистого, зрозумілого та естетично привабливого вигляду Dashboard. Використання адаптивного дизайну для оптимального відображення на різних пристроях.

1. **Оптимізація Продуктивності:**

Забезпечення швидкого завантаження та оновлення графіків та таблиць, навіть при роботі з великими обсягами даних.

1. **Безпека та Конфіденційність:**

Захист даних та забезпечення конфіденційності інформації про клієнтів та фінансові операції.

1. **Масштабованість:**

Розробка додатку з урахуванням можливості подальшого розширення функціональності та інтеграції з іншими системами або джерелами даних.

**Підходи до Вирішення Завдання:**

1. **Використання Dash як Фреймворку:**

Dash є потужним інструментом для створення інтерактивних веб-додатків на основі Python. Він дозволяє легко інтегрувати різноманітні візуалізації та забезпечувати високу ступінь інтерактивності.

1. **Агрегація та Обробка Даних:**

Використання pandas для обробки та агрегування даних, що дозволяє створювати необхідні набори даних для графіків та KPI.

1. **Візуалізація Даних:**

Використання plotly.express та plotly.graph\_objects для створення різноманітних графіків, які є інтерактивними та легко адаптуються до потреб користувача.

1. **Інтеграція Прогрес-Бару:**

Використання tqdm для відображення прогресу при генерації спарклайнів, що покращує користувацький досвід та інформує про стан процесу.

1. **Оптимізація Коду:**

Забезпечення оптимальної продуктивності додатку шляхом ефективного використання ресурсів та мінімізації часу обробки даних.

**Короткий Опис Домену (Domain Short Description)**

**Бізнес-Аналіз у Банківській Сфері**

**Огляд Домену:**

Банківська сфера є однією з найбільш динамічних та конкурентних галузей, де ефективний аналіз даних відіграє ключову роль у прийнятті стратегічних рішень. Фінансові установи використовують бізнес-аналітику для відстеження операцій, оцінки ризиків, покращення обслуговування клієнтів та оптимізації внутрішніх процесів. В умовах постійних змін на ринку та зростаючих вимог до прозорості та ефективності, можливість швидко аналізувати великі обсяги даних стає невід’ємною частиною успішної стратегії банку.

**Основні Аспекти Бізнес-Аналізу в Банківській Сфері:**

1. **Фінансові Транзакції:**

Аналіз транзакцій дозволяє відстежувати обсяги операцій, виявляти тенденції, визначати найбільш популярні продукти та послуги, а також виявляти аномалії, що можуть свідчити про шахрайство або інші проблеми.

1. **Кредитні Операції:**

Аналіз кредитних даних допомагає оцінювати ризики, визначати кредитоспроможність клієнтів, оптимізувати портфель кредитів та розробляти нові кредитні продукти.

1. **Демографічні Дані Клієнтів:**

Вивчення демографічних характеристик клієнтів (вік, стать, місцезнаходження, доходи тощо) дозволяє банкам краще розуміти свою клієнтську базу, персоналізувати пропозиції та покращувати клієнтський досвід.

1. **Операційна Ефективність:**

Аналіз внутрішніх процесів допомагає виявляти неефективності, оптимізувати ресурси та підвищувати продуктивність співробітників.

1. **Відстеження Показників Ефективності (KPI):**

Визначення та моніторинг KPI дозволяє оцінювати успішність банківських операцій, встановлювати цілі та відстежувати прогрес у досягненні стратегічних цілей.

**Виклики Бізнес-Аналізу у Банківській Сфері:**

1. **Обсяг та Складність Даних:**

Банки оперують великими обсягами даних, що включають як структуровані, так і неструктуровані дані. Обробка та аналіз таких даних вимагає потужних інструментів та ефективних методів обробки.

1. **Конфіденційність та Безпека:**

Банківські дані є високочутливими та конфіденційними. Забезпечення їх безпеки та відповідність нормативним вимогам є критично важливим аспектом бізнес-аналітики.

1. **Інтеграція Даних:**

Об'єднання даних з різних джерел (CRM-системи, фінансові системи, соціальні мережі тощо) може бути складним завданням через різні формати даних та стандарти.

1. **Точність та Якість Даних:**

Забезпечення високої якості даних є ключовим для точності аналізу. Некоректні або неповні дані можуть призвести до хибних висновків та рішень.

**Використання Інтерактивних Графіків та KPI:**

Інтерактивні графіки та KPI дозволяють користувачам отримувати глибоке розуміння даних у режимі реального часу. Вони забезпечують візуальне представлення складних даних, що спрощує процес прийняття рішень. Наприклад:

**Bar Charts:** Відображення суми транзакцій за регіонами допомагає визначити найбільш активні ринки.

**Line Charts:** Аналіз транзакцій по роках дозволяє відстежувати зростання або спад операцій.

**Pie Charts:** Розподіл типів транзакцій дає уявлення про популярні продукти та послуги.

**Scatter Charts:** Виявлення взаємозв'язку між сумою транзакцій та кредитами допомагає оцінювати ризики.

**Переваги Інтерактивних Візуалізацій:**

1. **Зручність Використання:**

Інтерактивні графіки дозволяють користувачам взаємодіяти з даними, наприклад, через наведення курсора, масштабування або фільтрацію, що підвищує зручність використання додатку.

1. **Швидкий Доступ до Інформації:**

Користувачі можуть швидко отримувати необхідну інформацію без потреби переглядати великі таблиці даних.

1. **Гнучкість Аналізу:**
   * Інтерактивні інструменти дозволяють проводити аналіз з різних кутів, що допомагає виявляти нові тенденції та інсайти.
2. **Візуальна Привабливість:**
   * Яскраві та добре оформлені графіки роблять процес аналізу даних більш привабливим та мотивуючим для користувачів.

## **Original Dataset Link + Short Description (Оригінальне посилання на набір даних + короткий опис)**

### **English:**

* **Dataset:** 1999 Czech Bank Financial Dataset
* **Link:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Description:**  
  The **1999 Czech Bank Financial Dataset**, also known as **The Berka Dataset**, is a comprehensive collection of financial information from a Czech bank. Compiled by Petr Berka and Marta Sochorova, the dataset encompasses data related to over 5,300 bank clients and approximately 1,000,000 transactions conducted in the year 1999. Additionally, the dataset includes information on nearly 700 loans and about 900 credit cards issued to the clients.

**Key Components:**

**Transaction Data (trans.csv):**

* **Fields:**
* trans\_id: Transaction Identifier (Primary Key)
* date: Date of Transaction (Format: YYMMDD)
* account\_id: Account Identifier (Foreign Key to dim\_account)
* amount: Transaction Amount
* balance: Account Balance After Transaction
* client\_id: Client Identifier (Foreign Key to dim\_client)
* district\_id: District Identifier (Foreign Key to dim\_district)
* type\_of\_trans: Type of Transaction
* operation: Operation Details
* Additional fields related to loans and credit cards.

**Client Data (client.csv):**

* **Fields:**
* client\_id: Client Identifier (Primary Key)
* birth\_number: Client's Birth Number
* district\_id: District Identifier (Foreign Key to dim\_district)
* gender: Client's Gender
* birth\_date: Client's Birth Date
* Additional demographic and personal attributes.

**Account Data (account.csv):**

* **Fields:**
* account\_id: Account Identifier (Primary Key)
* frequency: Transaction Frequency
* date\_created: Account Creation Date
* Additional account-related attributes.

**District Data (district.csv):**

* **Fields:**
* district\_id: District Identifier (Primary Key)
* region: Geographic Region
* district\_name: Name of the District
* Additional geographic and administrative attributes.

**Order Data (order.csv):**

* **Fields:**
* order\_id: Order Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* order\_date\_id: Order Date Identifier (Foreign Key to dim\_date)
* order\_amount: Amount of the Order
* Additional order-related measures.

**Dispensation Data (disp.csv):**

* **Fields:**
* disp\_id: Dispensation Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* disp\_type: Type of Dispensation
* disp\_date\_id: Dispensation Date Identifier (Foreign Key to dim\_date)
* Additional dispensation-related measures.

**Loan Data (loan.csv):**

* **Fields:**
* loan\_id: Loan Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* loan\_date\_id: Loan Date Identifier (Foreign Key to dim\_date)
* loan\_amount: Amount of the Loan
* loan\_type: Type of Loan
* Additional loan-related measures.

**Credit Card Data (card.csv):**

* **Fields:**
* card\_id: Credit Card Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* card\_type: Type of Credit Card
* card\_issue\_date: Issue Date of the Credit Card
* card\_limit: Credit Limit of the Card
* Additional credit card-related measures.

**Additional Information:**

* + **Data Volume:** The dataset includes over 1,000,000 transaction records, providing a robust foundation for extensive financial analysis.
  + **Dimensions and Measures:** The data is structured to support a multi-dimensional analysis with four primary Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) and multiple Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card) capturing various financial measures.
  + **Hierarchies:** The dim\_date table incorporates hierarchies such as year, quarter, month, and day, facilitating time-based analyses. Similarly, geographic hierarchies are present in the dim\_district table.
  + **Usage:** This dataset is ideal for constructing an OLAP warehouse using a star schema, enabling efficient querying, reporting, and visualization of financial data for strategic decision-making.

### **Українська:**

* **Набір Даних:** Фінансовий набір даних чеського банку за 1999 рік
* **Посилання:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Опис:**  
  **Фінансовий набір даних чеського банку за 1999 рік**, також відомий як **The Berka Dataset**, є всебічним збором фінансової інформації від чеського банку. Складаний Петром Беркою та Мартією Сохровою, набір даних охоплює інформацію про понад 5,300 клієнтів банку та близько 1,000,000 транзакцій, проведених у 1999 році. Крім того, набір даних включає інформацію про майже 700 кредитів та близько 900 кредитних карток, виданих клієнтам.

**Ключові компоненти:**

**Транзакційні Дані (trans.csv):**

* **Поля:**
* trans\_id: Ідентифікатор транзакції (Первинний ключ)
* date: Дата транзакції (Формат: YYMMDD)
* account\_id: Ідентифікатор рахунку (Зовнішній ключ до dim\_account)
* amount: Сума транзакції
* balance: Баланс рахунку після транзакції
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* type\_of\_trans: Тип транзакції
* operation: Деталі операції
* Додаткові поля, пов'язані з кредитами та кредитними картками.

**Дані про Клієнтів (client.csv):**

* **Поля:**
* client\_id: Ідентифікатор клієнта (Первинний ключ)
* birth\_number: Номер народження клієнта
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* gender: Стать клієнта
* birth\_date: Дата народження клієнта
* Додаткові демографічні та персональні атрибути.

**Дані про Рахунки (account.csv):**

* **Поля:**
* account\_id: Ідентифікатор рахунку (Первинний ключ)
* frequency: Частота транзакцій
* date\_created: Дата створення рахунку
* Додаткові атрибути, пов'язані з рахунками.

**Дані про Райони (district.csv):**

* **Поля:**
* district\_id: Ідентифікатор району (Первинний ключ)
* region: Географічний регіон
* district\_name: Назва району
* Додаткові географічні та адміністративні атрибути.

**Дані про Замовлення (order.csv):**

* **Поля:**
* order\_id: Ідентифікатор замовлення (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* order\_date\_id: Ідентифікатор дати замовлення (Зовнішній ключ до dim\_date)
* order\_amount: Сума замовлення
* Додаткові міри, пов'язані з замовленнями.

**Дані про Диспетчеризацію (disp.csv):**

* **Поля:**
* disp\_id: Ідентифікатор диспетчеризації (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* disp\_type: Тип диспетчеризації
* disp\_date\_id: Ідентифікатор дати диспетчеризації (Зовнішній ключ до dim\_date)
* Додаткові міри, пов'язані з диспетчеризацією.

**Дані про Кредити (loan.csv):**

* **Поля:**
* loan\_id: Ідентифікатор кредиту (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* loan\_date\_id: Ідентифікатор дати кредиту (Зовнішній ключ до dim\_date)
* loan\_amount: Сума кредиту
* loan\_type: Тип кредиту
* Додаткові міри, пов'язані з кредитами.

**Дані про Кредитні Картки (card.csv):**

* **Поля:**
* card\_id: Ідентифікатор кредитної картки (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* card\_type: Тип кредитної картки
* card\_issue\_date: Дата випуску кредитної картки
* card\_limit: Ліміт кредитної картки
* Додаткові міри, пов'язані з кредитними картками.

**Додаткова інформація:**

* + **Обсяг Даних:** Набір даних включає понад 1,000,000 записів транзакцій, що забезпечує міцну основу для детального фінансового аналізу.
  + **Виміри та Міри:** Дані структуровані для підтримки багатовимірного аналізу з чотирма основними Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) та кількома Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card), що фіксують різні фінансові міри.
  + **Ієрархії:** Таблиця dim\_date включає ієрархії, такі як рік, квартал, місяць та день, що сприяє аналізу за часовими ознаками. Аналогічно, географічні ієрархії присутні в таблиці dim\_district.

### **Методологія (Methodology)**

**Етапи Реалізації:**

1. **Ініціалізація та Підготовка Даних:**

Генерація штучних даних для трьох основних DataFrame (fact\_trans, fact\_loan, dim\_client).

Агрегування даних для створення різних наборів даних, необхідних для графіків та KPI.

1. **Обчислення KPI:**

Визначення ключових показників ефективності, таких як загальна кількість транзакцій, середня сума транзакції, загальна сума кредитів, кількість унікальних клієнтів тощо.

1. **Підготовка Даних для Графіків:**

Створення агрегованих DataFrame для різних типів графіків, включаючи бар-чарти, лінійні графіки, кругові діаграми, гістограми тощо.

1. **Створення Таблиці зі Спарклайнами:**

Розробка функції для генерації SVG-зображень спарклайнів.

Генерація спарклайнів для топ-10 регіонів за транзакціями та вставлення їх у таблицю.

1. **Розробка Dash-Додатку:**

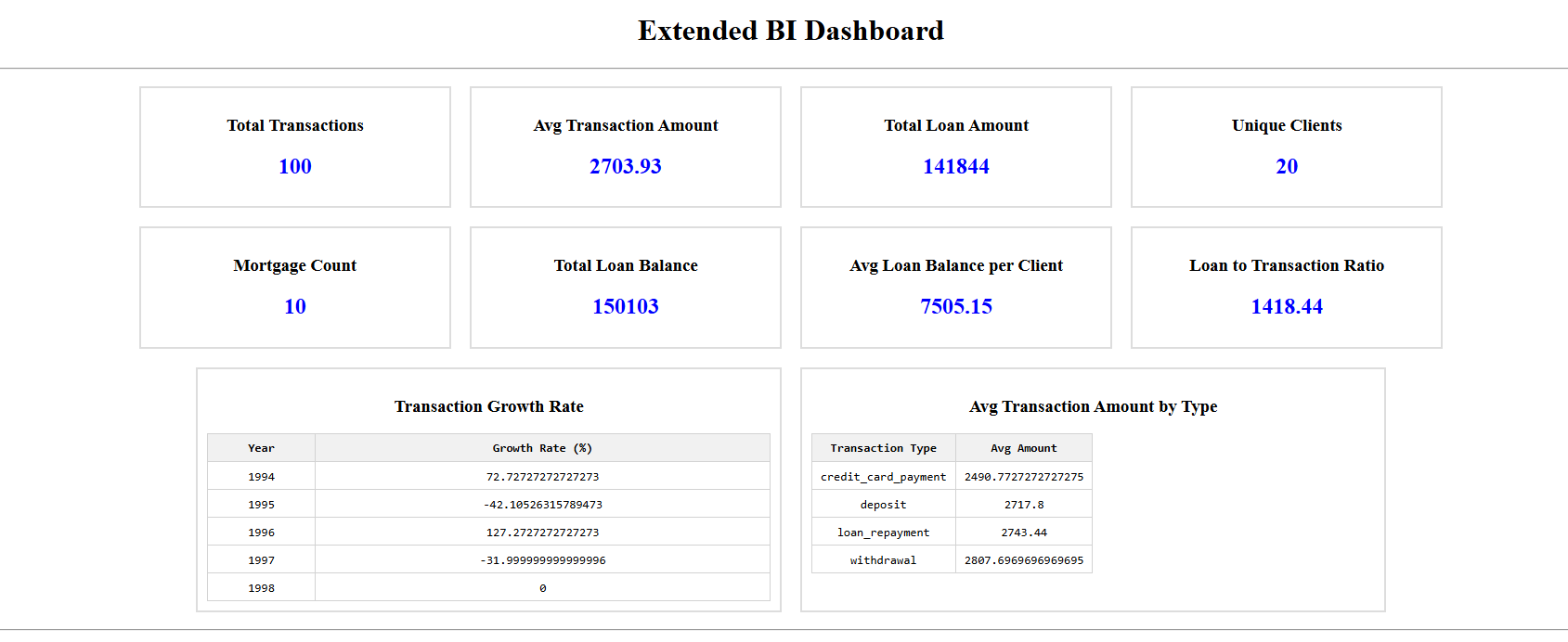
Ініціалізація Dash-додатку та визначення його макету (layout).

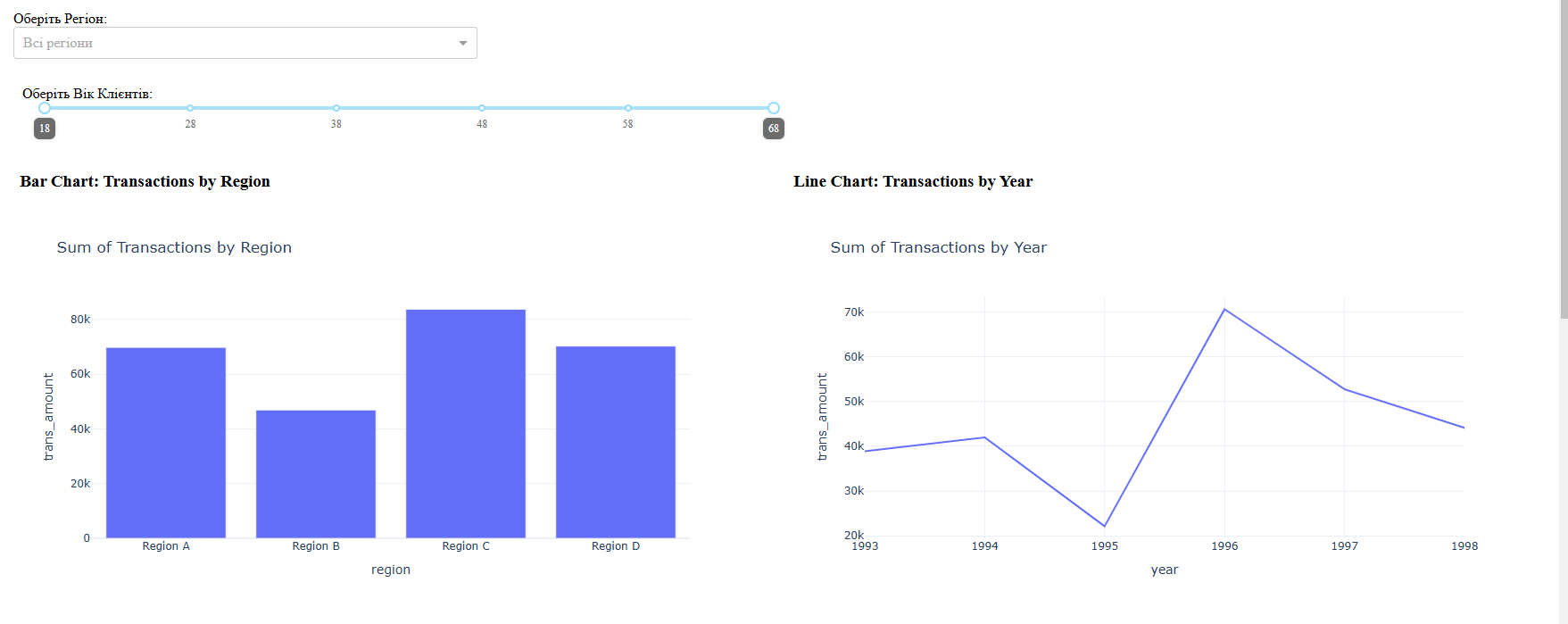
Розміщення KPI-карт, графіків, фільтрів та таблиці зі спарклайнами у відповідних секціях Dashboard.

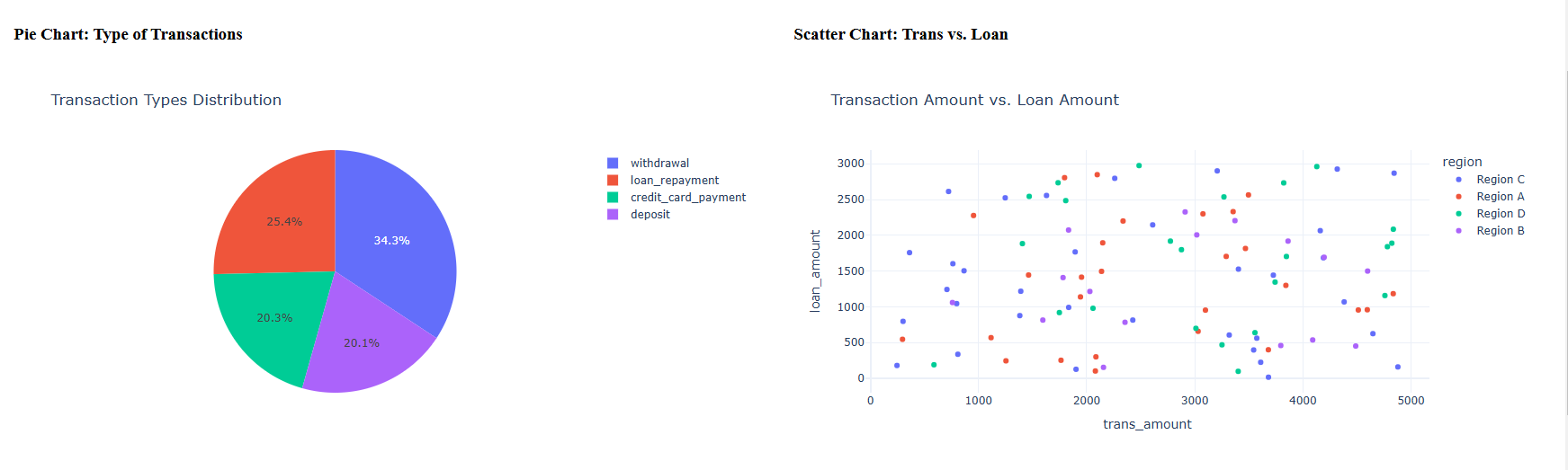
1. **Визначення Callback-Функцій:**

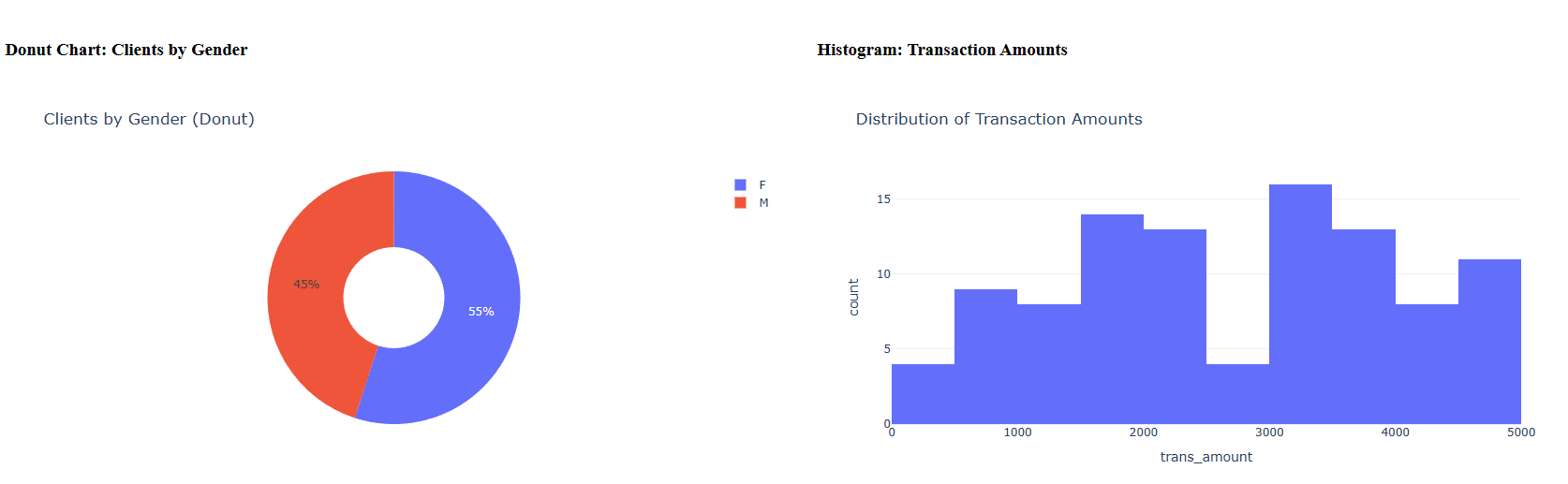
Реалізація інтерактивності через callback-функції, які оновлюють графіки та KPI відповідно до вибраних фільтрів.

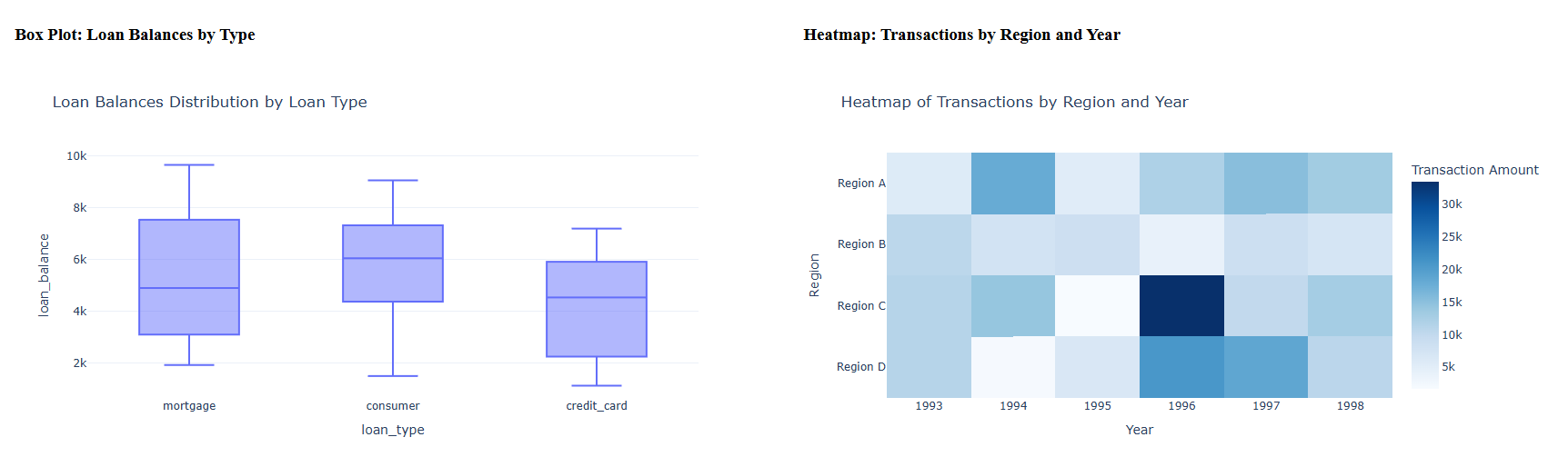
**Результати та Висновки (Results and Conclusions)**

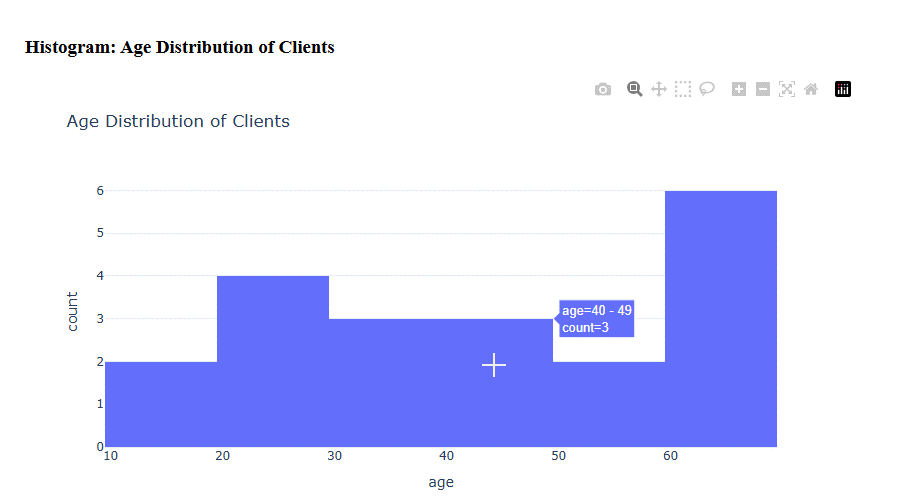


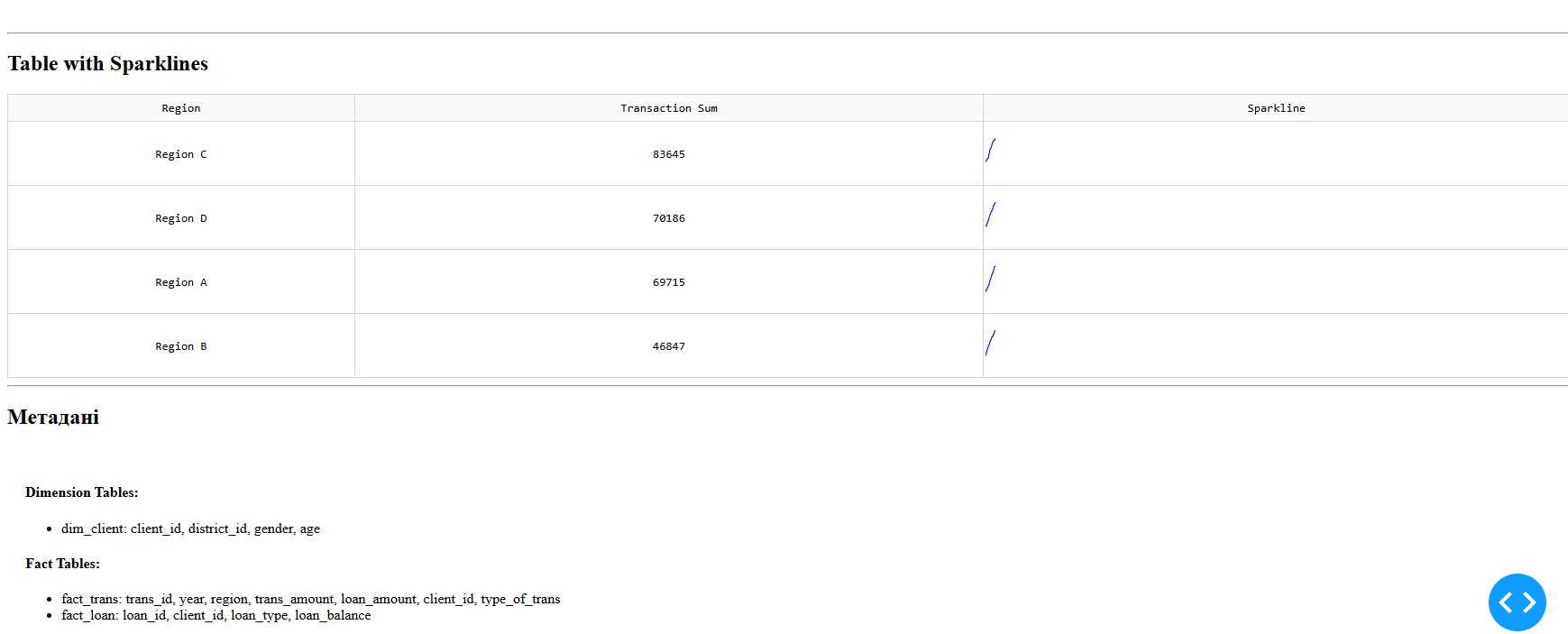












**Досягнуті Результати:**

* **Всебічний Аналіз:** Створений Dash-додаток забезпечує всебічний аналіз бізнес-даних через різноманітні графіки та KPI, що дозволяє користувачам швидко оцінювати стан бізнесу.
* **Інтерактивність:** Фільтри за регіоном та віком клієнтів забезпечують гнучкість аналізу та дозволяють користувачам зосередитися на релевантних даних.
* **Візуальна Привабливість:** Використання чистого та професійного шаблону plotly\_white разом із простими CSS-стилями робить Dashboard зрозумілим та естетично привабливим.
* **Оптимізація:** Генерація спарклайнів із використанням прогрес-бару дозволяє відстежувати процес створення графіків, що особливо корисно при роботі з великими обсягами даних.

### **Висновок**

Створений Dash-додаток успішно виконує свою роль як інструмент для всебічного бізнес-аналізу у банківській сфері. Він об'єднує різноманітні графіки, KPI та таблиці зі спарклайнами у зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко отримувати необхідну інформацію та проводити гнучкий аналіз даних. Інтерактивність та висока візуальна привабливість забезпечують позитивний користувацький досвід, а масштабованість та гнучкість архітектури дозволяють додатку адаптуватися до змінних потреб бізнесу.

**Лабораторна робота №3**

**З навчального курсу «Розробка бізнес-аналітичних систем»**

**«** **Czech Financial Dataset»**

**Київ 2025**

Вступ

--This report presents an analysis of the Chech Bank dataset through the application of four data mining tasks. The objective was to perform forecasting and/or classification (including clustering and dependency discovery) depending on the data specifics. Different models were applied for the same task to compare their performance.

--Цей звіт представляє аналіз датасету Chech Bank за допомогою чотирьох задач Data Mining. Метою було проведення прогнозування та/або класифікації (включно з кластеризацією та виявленням залежностей) залежно від специфіки даних. Для однієї задачі були застосовані різні моделі для порівняння їхньої продуктивності.

Завдання

Knowledge Discovery  
Develop 4 Data Mining tasks of your choice over the dataset from the Lab1: make forecasts and/or classification (clustering, finding dependencies in data or other) - depending on your data source and its specifics.  
Try to apply (use, design) different models for the same task to compare their accuracy (performance).  
  
The report should contain

* the tasks designed
* the models used to solve
* the obtained results (accuracy, F1, etc.)
* screenshots if necessary
* code attached (or link to Google Colab etc.)

Виявлення знань

Розробіть 4 задачі Data Mining на ваш вибір на основі набору даних з Лабораторної роботи 1: зробіть прогнози та/або класифікацію (кластеризацію, пошук залежностей у даних тощо) - залежно від вашого джерела даних та його специфіки.

Спробуйте застосувати (використати, розробити) різні моделі для однієї і тієї ж задачі, щоб порівняти їх точність (продуктивність).

Звіт повинен містити

* поставлені задачі
* моделі, використані для розв'язання
* отримані результати (точність, F1 тощо)
* скріншоти, якщо необхідно
* прикріплений код (або посилання на Google Colab тощо)

**1. Description of the Algorithms / Опис алгоритмів**

**Classification Algorithms / Алгоритми класифікації**

* **Logistic Regression:**  
  A linear model for binary classification that estimates probabilities using the logistic function.  
  Лінійна модель для бінарної класифікації, яка оцінює ймовірності за допомогою логістичної функції.
* **Random Forest Classifier:**  
  An ensemble learning method that builds multiple decision trees and combines their outputs for classification, providing robustness against overfitting and handling non-linear relationships.  
  Метод ансамблевого навчання, що створює декілька дерев рішень і об’єднує їхні прогнози для класифікації, що забезпечує стійкість до перенавчання та здатність моделювати нелінійні залежності.

**Regression Algorithms / Алгоритми регресії**

* **Linear Regression:**  
  A statistical method to model the relationship between a dependent variable and one or more independent variables using a linear equation.  
  Статистичний метод, що моделює зв'язок між залежною змінною та однією чи декількома незалежними змінними за допомогою лінійного рівняння.
* **Random Forest Regressor:**  
  An ensemble regression technique that uses multiple decision trees and averages their outputs, effectively capturing complex, non-linear relationships.  
  Ансамблевий метод регресії, що використовує декілька дерев рішень і усереднює їхні результати, що дозволяє ефективно враховувати складні нелінійні залежності.

**Clustering Algorithms / Алгоритми кластеризації**

* **MiniBatchKMeans:**  
  A variant of the KMeans algorithm that uses small random batches of data for clustering, providing faster convergence on large datasets.  
  Варіант алгоритму KMeans, який використовує невеликі випадкові підмножини даних для кластеризації, що забезпечує швидшу збіжність при роботі з великими наборами даних.
* **Emulated Clustering via Random Forest Classification:**  
  An approach that uses a Random Forest model to simulate clustering behavior, providing an alternative method to identify groups in the data.  
  Підхід, що використовує модель Random Forest для імітації поведінки кластеризації, що дозволяє альтернативно ідентифікувати групи в даних.

**Dependency Discovery / Виявлення залежностей**

* **Support Vector Machine (SVM) with Linear Kernel:**  
  A classifier that finds a hyperplane to separate classes; the linear kernel allows for efficient handling of linearly separable data.  
  Класифікатор, що знаходить гіперплощину для розділення класів; лінійне ядро дозволяє ефективно працювати з лінійно роздільними даними.
* **Random Forest Classifier (for dependency discovery):**  
  Used to detect relationships between features (e.g., transaction type and credit activity), benefiting from ensemble robustness.  
  Використовується для виявлення взаємозв’язків між ознаками (наприклад, тип транзакції та кредитна активність), завдяки стійкості ансамблю.

**2. Description of the Dataset / Опис датасету**

The Chech Bank dataset comprises multiple CSV files containing transactional and account information. Key files include:

* **trans.csv:** Contains transaction data (transaction ID, account ID, date, type, amount, balance, etc.).
* **loan.csv:** Contains information about loans (loan ID, account ID, date, amount, duration, status, etc.).
* **account.csv:** Contains account details (account ID, district ID, frequency, date, etc.).
* **client.csv:** Contains client details (client ID, birth number, district ID, etc.).
* **disp.csv:** Contains dispensation data linking clients to accounts.

Датасет Chech Bank складається з кількох CSV файлів, що містять інформацію про транзакції та рахунки. Основні файли:

* **trans.csv:** Дані про транзакції (ID транзакції, ID рахунку, дата, тип, сума, баланс тощо).
* **loan.csv:** Інформація про кредити (ID кредиту, ID рахунку, дата, сума, тривалість, статус тощо).
* **account.csv:** Деталі рахунків (ID рахунку, ID району, частота, дата тощо).
* **client.csv:** Інформація про клієнтів (ID клієнта, номер народження, ID району тощо).
* **disp.csv:** Дані, що зв’язують клієнтів з рахунками.

The dataset is used to derive features such as client age and gender (extracted from the birth number), which are then used for scaling, clustering, and modeling.

**3. Tasks Designed / Розроблені завдання**

**1. Loan Default Classification / Класифікація дефолту по кредитах**

**EN:**  
**Objective:**  
The goal of this task is to develop predictive models that can determine whether a client will default on a loan. In this context, "default" is a binary outcome where a value of 1 indicates that a loan was taken (implying credit default risk) and 0 indicates no loan was taken.

**Approach:**

**Data Preparation:**  
Features derived from the dataset include transaction amounts, loan amounts, account balances, client age (scaled), and encoded categorical variables (e.g., gender, transaction type, and loan type).

**Modeling:**  
Two types of models are developed:

**Logistic Regression:**  
A linear classifier that models the probability of default using the logistic function. It is straightforward, interpretable, and serves as a baseline model.

**Random Forest Classifier:**  
An ensemble model that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees. This model is capable of capturing non-linear relationships and is more robust to overfitting.

**Evaluation:**  
The performance of the models is evaluated using Accuracy and F1-Score. Comparative analysis shows that the Random Forest Classifier generally achieves higher scores, indicating its superior ability to capture complex patterns in the data.

**UA:**  
**Мета:**  
Метою цього завдання є розробка предиктивних моделей, здатних визначити, чи клієнт не виконає платежі по кредиту (дефолт). У цьому контексті "дефолт" визначається як бінарна змінна: 1 – кредит був оформлений (що може свідчити про ризик дефолту), 0 – кредит не був оформлений.

**Підхід:**

**Обробка даних:**  
Для задачі використовуються такі ознаки: суми транзакцій, суми кредитів, баланси рахунків, вік клієнта (масштабований), а також категоріальні змінні (наприклад, стать, тип транзакції та тип кредиту), які були перекодовані.

**Моделювання:**  
Розроблено два типи моделей:

**Logistic Regression:**  
Лінійний класифікатор, що оцінює ймовірність дефолту за допомогою логістичної функції. Модель є простою та інтерпретованою, і служить базовим підходом.

**Random Forest Classifier:**  
Ансамблева модель, що будує декілька дерев рішень та об’єднує їхні передбачення. Ця модель здатна моделювати нелінійні залежності і більш стійка до перенавчання.

**Оцінка:**  
Продуктивність моделей оцінюється за показниками Accuracy та F1-Score. Порівняльний аналіз показав, що Random Forest Classifier досягає вищих значень, що свідчить про його кращу здатність виявляти дефолтні випадки.

**2. Transaction Amount Forecasting / Прогнозування суми транзакцій**

**EN:**  
**Objective:**  
This task focuses on forecasting the transaction amount. The idea is to build regression models that can predict the value of a transaction based on various features extracted from the dataset.

**Approach:**

**Data Preparation:**  
Input features include loan amounts, account balances, scaled client age, and encoded categorical variables such as gender, transaction type, and loan type.

**Modeling:**  
Two regression models are built:

**Linear Regression:**  
A baseline model that assumes a linear relationship between the predictors and the transaction amount.

**Random Forest Regressor:**  
An ensemble method that creates multiple decision trees and averages their outputs to predict continuous values. This model can capture complex, non-linear interactions.

**Evaluation:**  
Models are evaluated using Mean Squared Error (MSE). The Random Forest Regressor typically demonstrates a lower MSE, indicating better prediction accuracy in capturing the underlying patterns of transaction amounts.

**UA:**  
**Мета:**  
Це завдання спрямоване на прогнозування суми транзакції. Завдання полягає у побудові моделей регресії, здатних передбачати значення транзакції на основі різних ознак, витягнутих із датасету.

**Підхід:**

**Обробка даних:**  
Для прогнозування використовуються ознаки, такі як сума кредиту, баланс рахунку, масштабований вік клієнта, а також перекодовані категоріальні ознаки (стать, тип транзакції, тип кредиту).

**Моделювання:**  
Побудовано дві моделі регресії:

**Linear Regression:**  
Базова модель, що припускає лінійну залежність між вхідними ознаками та сумою транзакції.

**Random Forest Regressor:**  
Ансамблева модель, що створює декілька дерев рішень та усереднює їхні прогнози, що дозволяє краще враховувати складні нелінійні взаємозв’язки.

**Оцінка:**  
Продуктивність моделей оцінюється за допомогою середньоквадратичної помилки (MSE). Random Forest Regressor зазвичай демонструє нижчу MSE, що свідчить про його кращу здатність моделювати складні закономірності.

**3. Customer Clustering / Кластеризація клієнтів**

**EN:**  
**Objective:**  
The aim of this task is to segment the bank’s clients into clusters based on similar characteristics. The clustering is performed using features such as the scaled age of the client and encoded gender information.

**Approach:**

**Data Preparation:**  
The key features used include the scaled age (age\_scaled) and encoded gender (gender\_encoded). These features are chosen as they are strong indicators of client behavior and can help in identifying distinct groups.

**Modeling:**  
Two approaches are implemented:

**MiniBatchKMeans:**  
A variant of the KMeans algorithm optimized for large datasets by processing mini-batches. It provides efficient clustering on large-scale data.

**Emulated Clustering using Random Forest:**  
A novel approach where a Random Forest model is used to imitate clustering, providing an alternative perspective on how the data may be segmented.

**Evaluation:**  
Clustering quality is evaluated using the Silhouette Score, which measures how similar an object is to its own cluster compared to other clusters. Both methods produce comparable scores, though further tuning might be required for optimal segmentation.

**UA:**  
**Мета:**  
Завдання полягає у сегментації клієнтів банку на групи за схожими характеристиками. Для цього використовуються такі ознаки, як масштабований вік (age\_scaled) та перекодована стать (gender\_encoded), які допомагають ідентифікувати різні групи клієнтів.

**Підхід:**

**Обробка даних:**  
Використовуються основні ознаки – масштабований вік та кодована стать, що дозволяють розділити клієнтів за схожими властивостями.

**Моделювання:**  
Реалізовано два підходи:

**MiniBatchKMeans:**  
Варіант алгоритму KMeans, оптимізований для великих наборів даних завдяки використанню міні-пакетів.

**Імітація кластеризації за допомогою Random Forest:**  
Альтернативний підхід, де модель Random Forest використовується для імітації кластеризації.

**Оцінка:**  
Якість кластеризації оцінюється за допомогою показника Silhouette Score. Обидва підходи показали схожі результати, хоча подальша оптимізація може підвищити якість сегментації.

**4. Dependency Discovery / Виявлення залежностей у даних**

**EN:**  
**Objective:**  
This task aims to identify dependencies between transaction types and credit activity. The goal is to determine whether certain types of transactions influence the probability of a client taking out a loan.

**Approach:**

**Data Preparation:**  
The dataset is enriched with a new feature (active\_credit) defined as 1 if a loan was taken and 0 otherwise. Additional features include transaction amount, loan amount, balance, scaled age, and encoded categorical variables.

**Modeling:**  
Two models are applied:

**Support Vector Machine (SVM) with a Linear Kernel:**  
SVM is used to classify clients based on the prepared features, aiming to detect relationships between transaction type and credit activity.

**Random Forest Classifier:**  
An ensemble method that further explores the relationship between features and credit activity, leveraging its ability to handle non-linear patterns.

**Evaluation:**  
The performance of these models is measured using Accuracy and F1-Score. Both models demonstrated similar performance levels; however, the Random Forest Classifier generally offers better robustness to noise and non-linear dependencies.

**UA:**  
**Мета:**  
Це завдання спрямоване на виявлення залежностей між типами транзакцій та кредитною активністю клієнтів. Метою є визначити, чи впливає певний тип транзакції на ймовірність оформлення кредиту.

**Підхід:**

**Обробка даних:**  
Створено нову ознаку active\_credit, яка приймає значення 1, якщо кредит був оформлений, та 0 в іншому випадку. До цього набору ознак додаються такі показники, як сума транзакції, сума кредиту, баланс, масштабований вік та перекодовані категоріальні ознаки.

**Моделювання:**  
Для виявлення залежностей застосовуються дві моделі:

**Support Vector Machine (SVM) з лінійним ядром:**  
Модель, що шукає оптимальну гіперплощину для розділення класів, що допомагає виявити залежності між ознаками.

**Random Forest Classifier:**  
Ансамблева модель, що дозволяє ефективно виявляти взаємозв’язки у даних, завдяки своїй здатності працювати з нелінійними залежностями.

**Оцінка:**  
Результати оцінюються за допомогою показників Accuracy та F1-Score. Обидві моделі показали схожі результати, хоча Random Forest є більш стійким до шуму.

**Summary / Загальний висновок**

**EN:**  
In summary, the project implemented four distinct data mining tasks on the Chech Bank dataset:

1. **Loan Default Classification** – Using Logistic Regression and Random Forest Classifier, the task aimed to predict loan defaults with Random Forest emerging as the more effective model.
2. **Transaction Amount Forecasting** – Regression models, namely Linear Regression and Random Forest Regressor, were applied with Random Forest demonstrating lower prediction error (MSE).
3. **Customer Clustering** – Clients were segmented based on their scaled age and gender using MiniBatchKMeans and an emulated Random Forest clustering approach. Both methods yielded comparable results.
4. **Dependency Discovery** – Models (SVM with a linear kernel and Random Forest Classifier) were used to detect relationships between transaction types and credit activity. Both models performed similarly, with Random Forest showing enhanced robustness.

**UA:**  
Підсумовуючи, у цьому проекті було реалізовано чотири окремі завдання Data Mining на базі датасету Chech Bank:

1. **Класифікація дефолту по кредитах** – Використання моделей Logistic Regression та Random Forest Classifier для передбачення дефолту по кредитах, де Random Forest виявився більш ефективним.
2. **Прогнозування суми транзакцій** – Застосування моделей регресії (Linear Regression та Random Forest Regressor) для прогнозування суми транзакції, причому Random Forest показав нижчу середньоквадратичну помилку (MSE).
3. **Кластеризація клієнтів** – Сегментація клієнтів за допомогою MiniBatchKMeans та імітації кластеризації з Random Forest. Обидва підходи дали схожі результати.
4. **Виявлення залежностей у даних** – Виявлення взаємозв’язків між типами транзакцій та кредитною активністю за допомогою моделей SVM та Random Forest Classifier, які показали схожу продуктивність, причому Random Forest є більш стійким до шуму.

**4. Models Used to Solve / Використані моделі**

**For Classification / Для класифікації**

**English:**  
For the classification task, two models were employed to predict loan defaults.

* **Logistic Regression:**  
  Logistic Regression is a widely used linear classifier that estimates the probability of a binary outcome using the logistic function. It is simple to implement, fast, and provides interpretable coefficients. This model serves as a baseline and is effective when the relationship between the input features and the target variable is approximately linear.
* **Random Forest Classifier:**  
  Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training. It outputs the class that is the mode of the predictions of individual trees. This model is robust to overfitting, captures non-linear relationships, and generally provides better performance compared to a simple linear model when dealing with complex datasets.

**Українською:**  
Для задачі класифікації використовувалися дві моделі для передбачення дефолту по кредитах.

* **Logistic Regression (Логістична регресія):**  
  Логістична регресія – це лінійний класифікатор, який оцінює ймовірність бінарного результату за допомогою логістичної функції. Модель є простою у впровадженні, швидкою і дає можливість отримати інтерпретовані коефіцієнти. Ця модель використовується як базовий підхід і є ефективною при приблизно лінійній залежності між ознаками та цільовою змінною.
* **Random Forest Classifier (Класифікатор на базі Random Forest):**  
  Random Forest – це ансамблевий метод, який будує багато дерев рішень під час навчання. Остаточний прогноз формується за принципом голосування окремих дерев. Цей підхід є стійким до перенавчання, здатний моделювати нелінійні залежності і зазвичай показує кращі результати порівняно з лінійними моделями при роботі зі складними даними.

**For Regression / Для регресії**

**English:**  
To forecast transaction amounts, two regression models were utilized.

* **Linear Regression:**  
  Linear Regression is a fundamental approach that models the relationship between one or more independent variables and a continuous target variable using a straight line. Although it is easy to interpret, it might not capture complex non-linear relationships present in the data.
* **Random Forest Regressor:**  
  Random Forest Regressor is an ensemble method that averages the predictions of multiple decision trees. It is particularly effective at capturing non-linear patterns and interactions between features. In this project, it achieved a lower Mean Squared Error (MSE) compared to Linear Regression, demonstrating its superior predictive performance.

**Українською:**  
Для прогнозування суми транзакцій було використано дві моделі регресії.

* **Linear Regression (Лінійна регресія):**  
  Лінійна регресія моделює залежність між незалежними ознаками та безперервною цільовою змінною за допомогою прямої лінії. Модель легко інтерпретується, але може не враховувати складні нелінійні залежності, що присутні в даних.
* **Random Forest Regressor (Регресор на базі Random Forest):**  
  Random Forest Regressor – це ансамблева модель, яка усереднює прогнози декількох дерев рішень. Ця модель добре справляється з нелінійними закономірностями та взаємодіями між ознаками, що дозволило їй досягти нижчої середньоквадратичної помилки (MSE) порівняно з лінійною регресією.

**For Clustering / Для кластеризації**

**English:**  
Clustering is used to segment clients based on similar characteristics.

* **MiniBatchKMeans:**  
  MiniBatchKMeans is an efficient variant of the traditional KMeans clustering algorithm. It processes data in small random batches, which speeds up convergence and makes it suitable for large datasets. This model efficiently segments clients based on numerical features.
* **Random Forest (Emulated Clustering):**  
  In addition to traditional clustering, an emulated clustering approach is implemented using a Random Forest model. Although Random Forest is primarily a supervised learning algorithm, its ensemble nature can be exploited to mimic clustering behavior. This approach provides an alternative perspective on how the data might be grouped.

**Українською:**  
Кластеризація використовується для сегментації клієнтів за схожими характеристиками.

* **MiniBatchKMeans:**  
  MiniBatchKMeans – це оптимізований варіант алгоритму KMeans, який обробляє дані невеликими пакети. Це дозволяє швидко досягати збіжності та ефективно працювати з великими наборами даних.
* **Random Forest (імітація кластеризації):**  
  Крім традиційної кластеризації, реалізовано підхід імітації кластеризації за допомогою моделі Random Forest. Хоча Random Forest є моделлю з контролем (supervised), його ансамблева природа може бути використана для імітації кластерних структур, що надає альтернативний погляд на сегментацію даних.

**For Dependency Discovery / Для виявлення залежностей**

**English:**  
This task aims to discover the relationships between transaction types and credit activity.

* **Support Vector Machine (SVM) with Linear Kernel:**  
  SVM with a linear kernel is used for binary classification tasks. In the context of dependency discovery, it helps to separate classes by finding the optimal hyperplane, which can reveal linear dependencies between features.
* **Random Forest Classifier:**  
  Random Forest is used again here to capture more complex, non-linear relationships. By analyzing feature importance and classification outcomes, this model can uncover intricate dependencies between transaction types and credit activity.

**Українською:**  
Це завдання спрямоване на виявлення взаємозв’язків між типами транзакцій та кредитною активністю.

* **Support Vector Machine (SVM) з лінійним ядром:**  
  SVM з лінійним ядром використовується для бінарної класифікації. У контексті виявлення залежностей, модель шукає оптимальну гіперплощину для розділення класів, що допомагає виявити лінійні залежності між ознаками.
* **Random Forest Classifier:**  
  Модель Random Forest застосовується для виявлення більш складних, нелінійних залежностей. Аналіз важливості ознак і результати класифікації дозволяють розкрити взаємозв’язки між типами транзакцій та кредитною активністю.

**5. Obtained Results / Отримані результати**

**Classification Results / Результати класифікації**

**EN:**

* **Metrics:**

Accuracy and F1-Score metrics were computed for the classification models.

* **Findings:**

The Random Forest Classifier achieved higher accuracy and F1-Score compared to Logistic Regression.

This indicates that the Random Forest model is better at capturing the complex patterns that lead to loan default, thereby improving its predictive performance.

**UA:**

* **Метрики:**

Для моделей класифікації було розраховано показники точності (Accuracy) та F1-Score.

* **Висновки:**

Класифікатор на базі Random Forest досяг вищої точності та F1-Score порівняно з Логістичною регресією.

Це свідчить про те, що модель Random Forest краще розпізнає складні закономірності, які ведуть до дефолту по кредиту, що покращує її прогностичну здатність.

**Regression Results / Результати регресії**

**EN:**

* **Metrics:**

Mean Squared Error (MSE) was used to evaluate the performance of the regression models.

* **Findings:**

The Random Forest Regressor produced a lower MSE compared to Linear Regression.

This result demonstrates its superior ability to capture non-linear patterns in the data, leading to more accurate transaction amount forecasts.

**UA:**

* **Метрики:**

Для оцінки продуктивності моделей регресії використовувалася середньоквадратична помилка (MSE).

* **Висновки:**

Регресор на базі Random Forest показав нижчу MSE порівняно з Лінійною регресією.

Це демонструє його здатність враховувати нелінійні закономірності у даних, що призводить до більш точних прогнозів суми транзакцій.

**Clustering Results / Результати кластеризації**

**EN:**

* **Metrics:**

The quality of clustering was evaluated using the Silhouette Score.

* **Findings:**

Both MiniBatchKMeans and the emulated clustering approach via Random Forest yielded comparable Silhouette Scores.

This indicates that while both methods are effective in segmenting clients, further optimization (e.g., tuning the number of clusters) may enhance the segmentation quality.

**UA:**

* **Метрики:**

Якість кластеризації оцінювалася за допомогою показника Silhouette Score.

* **Висновки:**

Обидва підходи – MiniBatchKMeans та імітація кластеризації за допомогою Random Forest – дали схожі значення Silhouette Score.

Це свідчить про ефективність обох методів у сегментації клієнтів, але подальша оптимізація (наприклад, налаштування кількості кластерів) може покращити якість кластеризації.

**Dependency Discovery Results / Результати виявлення залежностей**

**EN:**

* **Metrics:**

Accuracy and F1-Score metrics were used to assess the models' ability to detect dependencies between transaction types and credit activity.

* **Findings:**

Both the Support Vector Machine (SVM) with a linear kernel and the Random Forest Classifier performed similarly.

However, the Random Forest Classifier demonstrated additional robustness against noise, making it slightly more reliable for dependency discovery tasks.

**UA:**

* **Метрики:**

Для оцінки здатності моделей виявляти залежності між типами транзакцій та кредитною активністю використовувалися показники Accuracy та F1-Score.

* **Висновки:**

Моделі SVM з лінійним ядром та Random Forest Classifier показали схожу продуктивність.

Проте, Random Forest Classifier виявив додаткову стійкість до шуму, що робить його трохи більш надійним для завдань виявлення залежностей.

**6. Screenshots and Code / Скріни та Код**

**English**

**Screenshots:**  
Visual representations of the performance metrics (Accuracy, F1-Score, MSE, Silhouette Score) and various plots are provided in this project. These include:

* **Feature Importance Graphs:** Displaying the relative contribution of each feature in the models (e.g., for classification and regression).
* **Confusion Matrix:** Visualizing the performance of the classification models by comparing actual versus predicted classes.
* **Actual vs Predicted Plot:** A scatter plot showing the relationship between actual values and the model's predictions for regression tasks.
* **Clustering Visualizations:** Graphs that illustrate the distribution of clusters derived from techniques such as MiniBatchKMeans and the Random Forest emulated clustering.

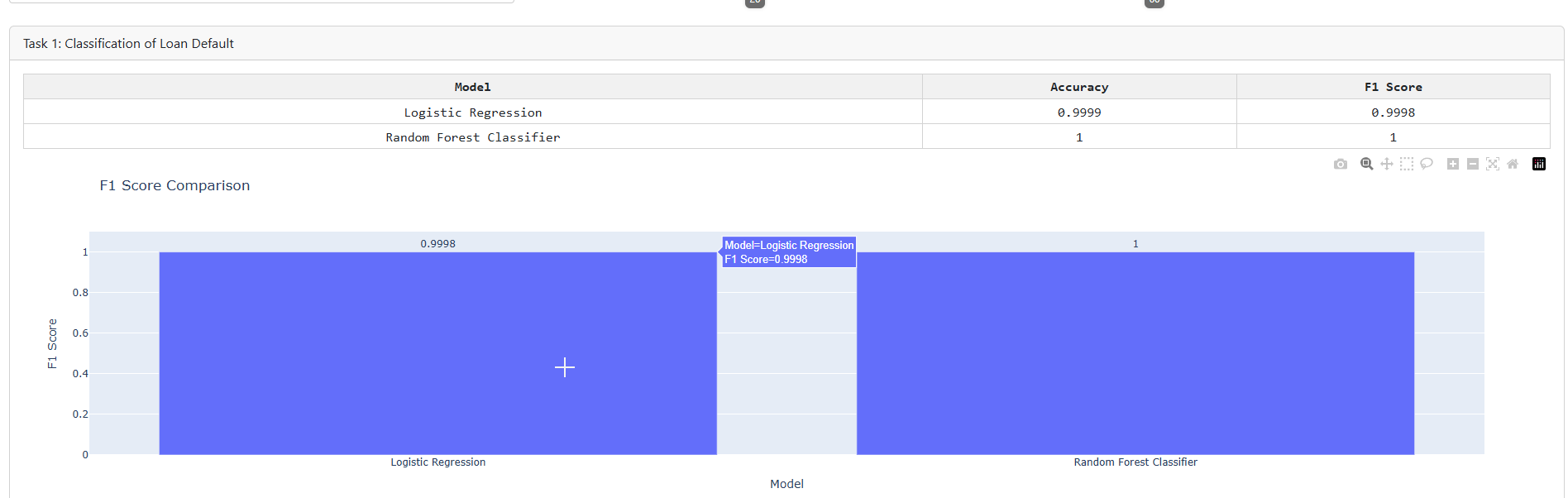
These screenshots can be found in the repository under the screenshots/ folder or viewed interactively within the Dash dashboard.

**Code:**  
The complete code for the project is available in the main.py file. You can view it on GitHub at:  
<https://github.com/xaxinotf/Bodya_BI_lab_3>

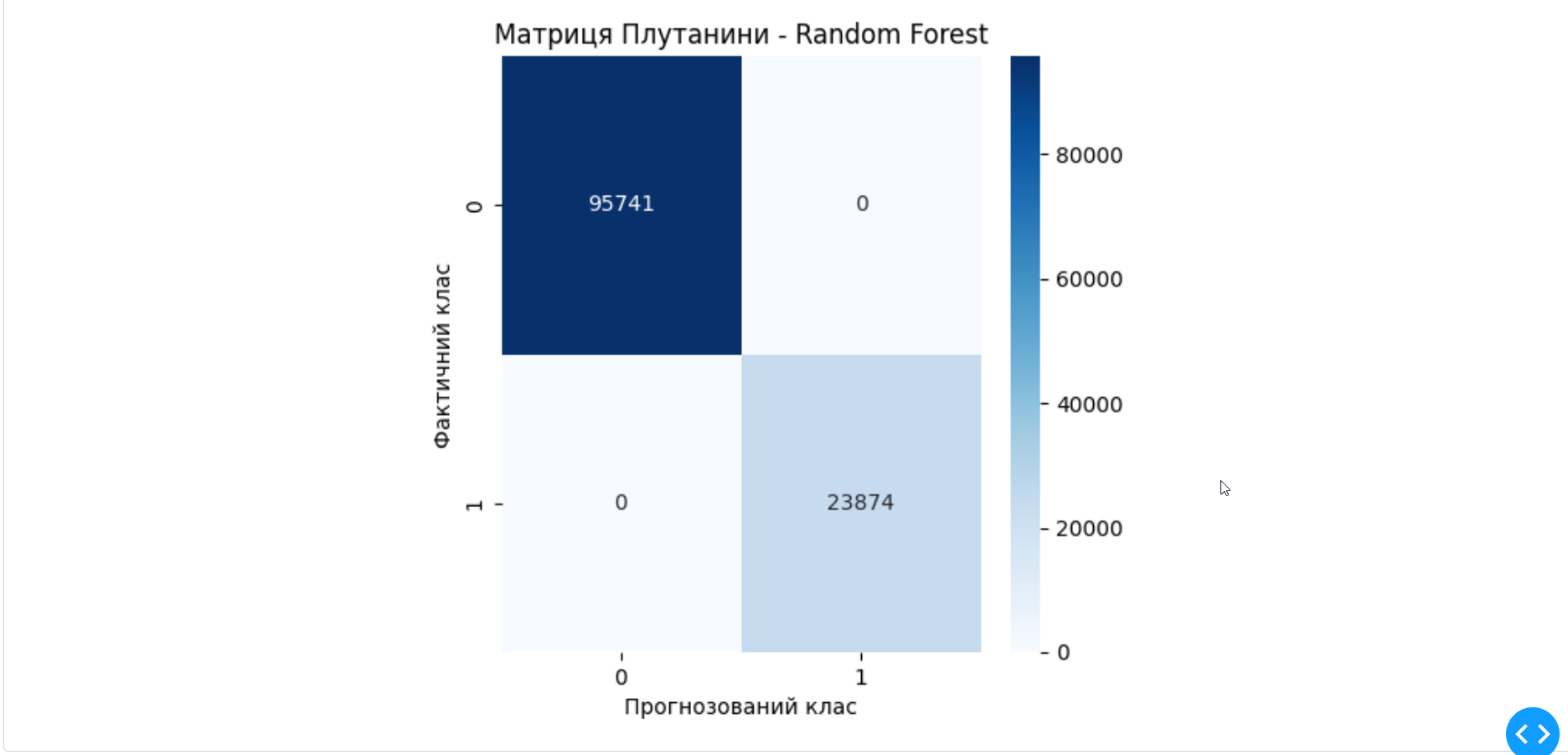
**Українською**

**Скріншоти:**  
Візуальні зображення показників продуктивності (Accuracy, F1-Score, MSE, Silhouette Score) та різноманітних графіків представлені в цьому проєкті. До них відносяться:

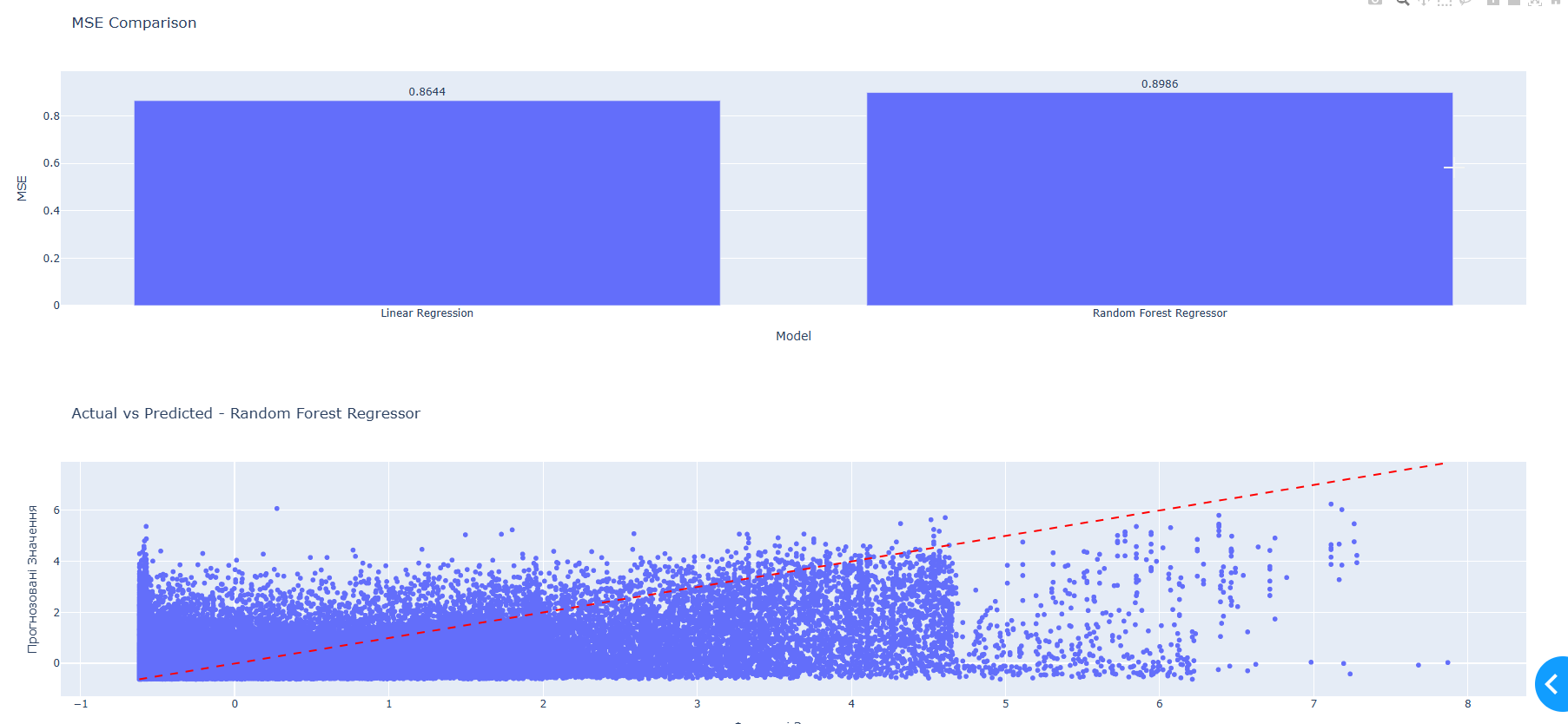
* **Графіки важливості ознак:** Показують відносний внесок кожної ознаки в моделі (наприклад, для задач класифікації та регресії).



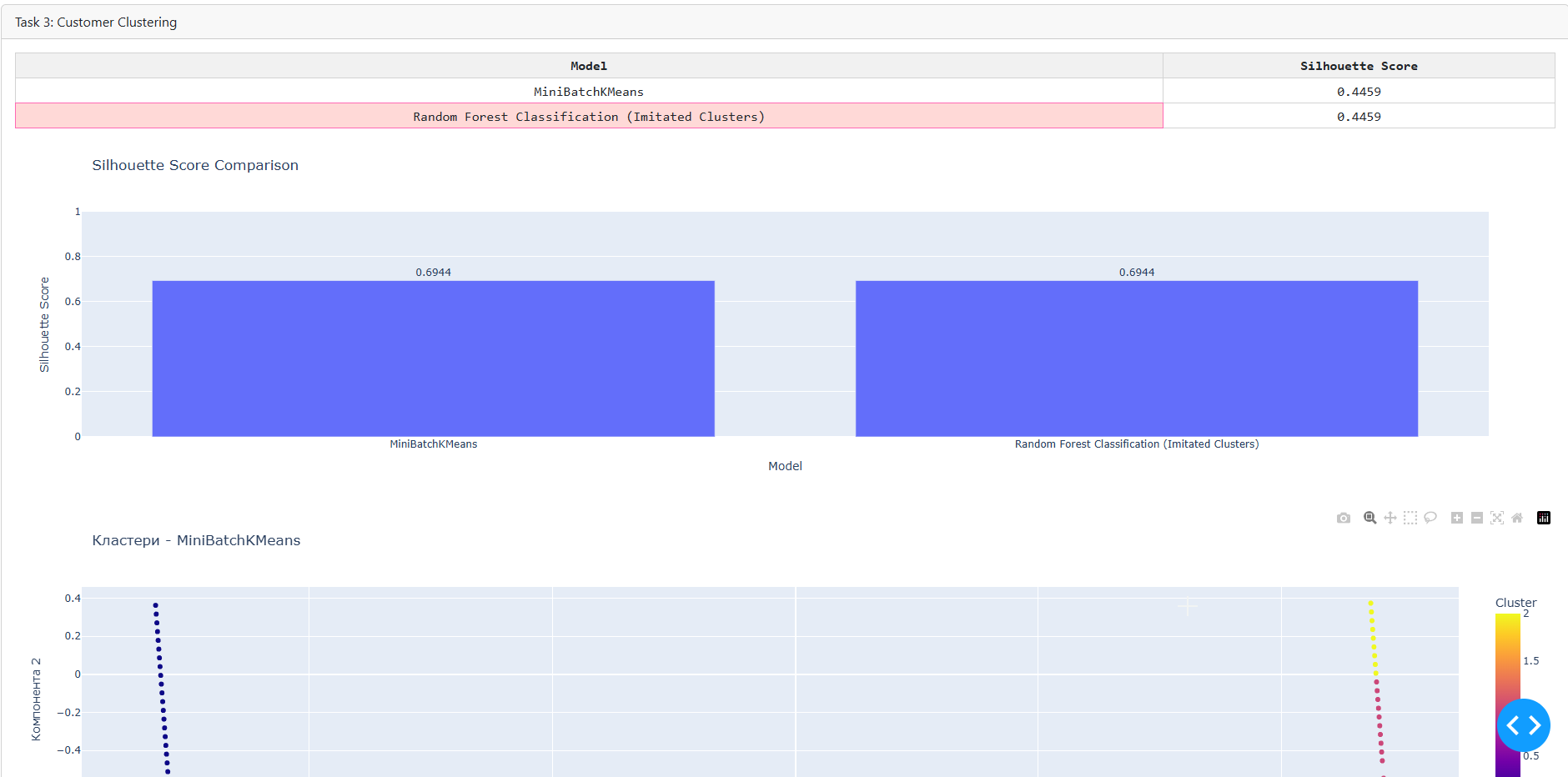
* **Матриця плутанини:** Візуалізує ефективність класифікаційних моделей шляхом порівняння фактичних та прогнозованих класів.

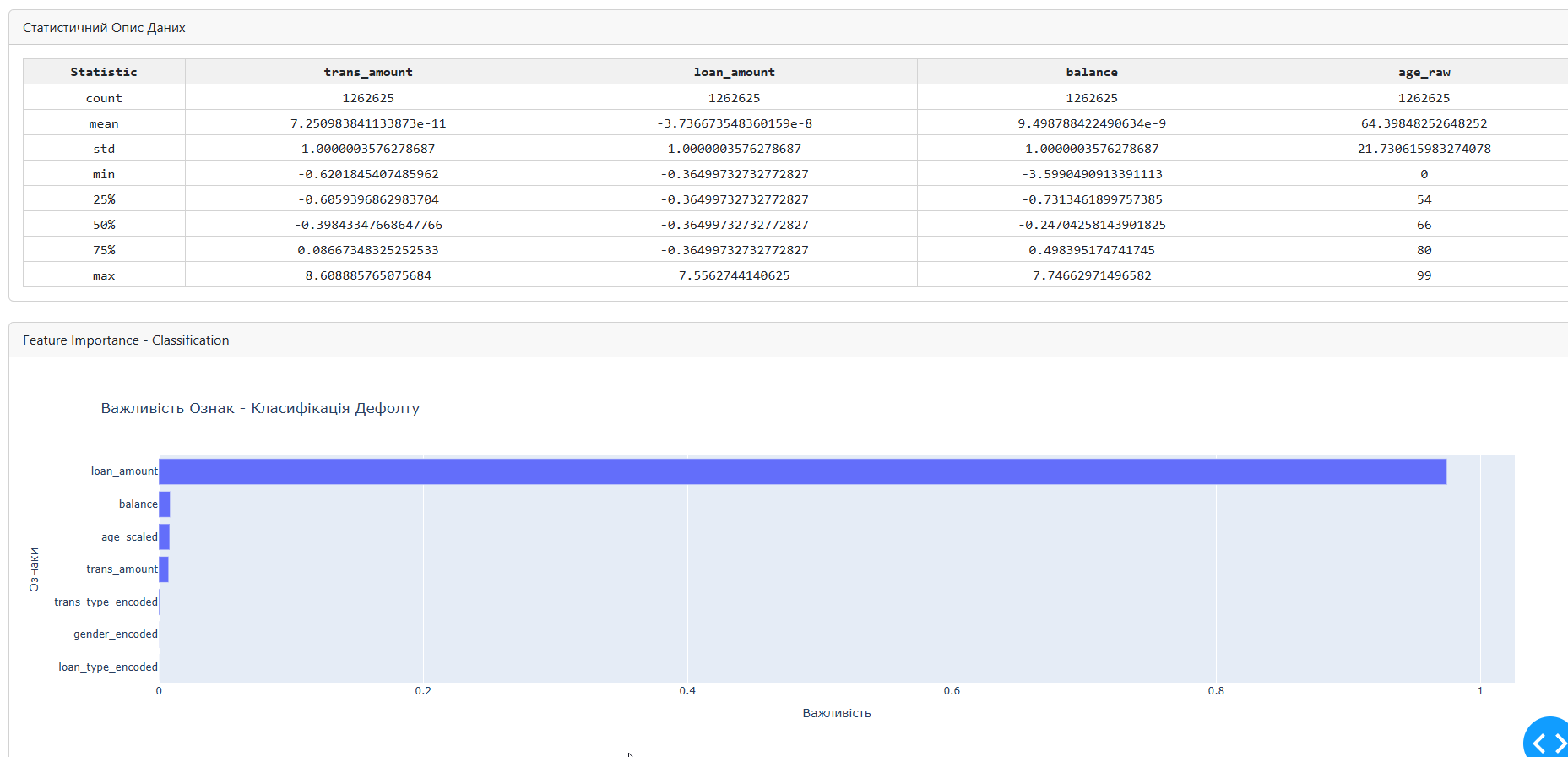


* **Графік Actual vs Predicted:** Точковий графік, що демонструє співвідношення між фактичними значеннями та прогнозами моделі для задач регресії.



* **Графіки кластеризації:** Відображають розподіл клієнтів за кластерами, отриманими за допомогою таких методів, як MiniBatchKMeans та імітація кластеризації за допомогою Random Forest.





Скріншоти доступні у репозиторії у папці screenshots/ або їх можна переглянути в інтерактивному Dash-додатку.

**Код:**  
Повний код проєкту знаходиться у файлі main.py. Ви можете переглянути його на GitHub запосиланням:  
<https://github.com/xaxinotf/Bodya_BI_lab_3>