

УДК: 330.322.012

Т. Патральський, студент групи ПКМ-51

Тернопільський національний технічний університет імені Івана Пулюя, Україна

Науковий керівник: Н.Гарматій, канд. екон. наук, доц.

ПРАКТИЧНІ АСПЕКТИ ВИКОРИСТАННЯ ЦИФРОВИХ ПЛАТФОРМ У DATA ANALYSIS

T. Patralskyi student PKM-51 group

Supervisor: N. Harmatiy, Ph.D.in Economics, Assos Prof.

Ternopil Ivan Puluj National Technical University, Ukraine

PRACTICAL ASPECTS OF USING DIGITAL PLATFORMS IN DATA ANALYSIS

У сучасній цифровій економіці, дані можуть бути представлені у різному форматі: таблиці, бази даних, графічний матеріал, закодовані файли та інші формати, в тому числі і цифрові. Для того щоб провести аналіз даних потрібно усе впорядкувати таким чином, щоб аналітик зміг провести аналіз чи сформулювати звіт для подальших дій. Одним із способів аналізу даних можна вважати візуальне представлення цих даних у формі інформаційної панелі, але для цього має пройти кілька процесів (див. рисунок 1).

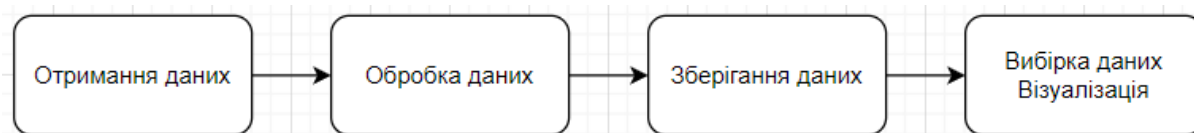


Рис.1. Етапи обробки даних для аналізу

Спочатку потрібно знайти джерело даних, які потрібно аналізувати, доступ до джерела даних може бути API(application programming interface) чи просте підключення до бази даних чи до будь-яких інших джерел, навіть таблиць.

Можна використати для вивантаження даних мову програмування Python та нескладні скрипти які дозволяють легко отримати дані які потрібно, також за допомогою скрипта можна ці дані упорядкувати, перетворити у правильні формати та встановити ієрарію для них (див. рисунок 2).

```
import pandas as pd
import re
# Read the CSV file into a DataFrame
df = pd.read_csv('C:/Users/t.patral'skiy/Downloads/table.csv', encoding='utf-8')

# Preprocess the data in t1 (equivalent to your first CTE)
t1 = df['platnuku'].str.lower().str.replace(' ', '').str.replace('.', '').str.replace('-', '').unique()
t1 = pd.Series(t1)

C:/Users/t.patral'skiy/AppData/Local/Temp/ipykernel_7208/2477461942.py:2: FutureWarning: The default value of regex will change from True to False in a future version.
t1 = df['platnuku'].str.lower().str.replace(' ', '').str.replace('.', '').str.replace('-', '').unique()
C:/Users/t.patral'skiy/AppData/Local/Temp/ipykernel_7208/2477461942.py:2: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
t1 = df['platnuku'].str.lower().str.replace(' ', '').str.replace('.', '').str.replace('-', '').unique()

# Preprocess the data in t2 (equivalent to your second CTE)
df['owner_dzk_short'] = df['owner_dzk'].str.split(' ').str[0] + ' ' + df['owner_dzk'].str.split(' ').str[1].str[0] + ' ' + df['owner_dzk'].str.split(' ').str[2].str[0]
df['owner_rrp_short'] = df['owner_rrp'].str.split(' ').str[0] + ' ' + df['owner_rrp'].str.split(' ').str[1].str[0] + ' ' + df['owner_rrp'].str.split(' ').str[2].str[0]
df['user_dzk_short'] = df['user_dzk'].str.split(' ').str[0] + ' ' + df['user_dzk'].str.split(' ').str[1].str[0] + ' ' + df['user_dzk'].str.split(' ').str[2].str[0]
df['user_rrp_short'] = df['user_rrp'].str.split(' ').str[0] + ' ' + df['user_rrp'].str.split(' ').str[1].str[0] + ' ' + df['user_rrp'].str.split(' ').str[2].str[0]

# Initialize a list to store the results
results = []

for platnuku in t1:
    condition = (
        (df['owner_dzk'].str.lower().str.contains(re.escape(platnuku)) | df['owner_dzk_short'].str.lower().str.contains(re.escape(platnuku))) |
        (df['owner_rrp'].str.lower().str.contains(re.escape(platnuku)) | df['owner_rrp_short'].str.lower().str.contains(re.escape(platnuku))) |
        (df['user_dzk'].str.lower().str.contains(re.escape(platnuku)) | df['user_dzk_short'].str.lower().str.contains(re.escape(platnuku))) |
        (df['user_rrp'].str.lower().str.contains(re.escape(platnuku)) | df['user_rrp_short'].str.lower().str.contains(re.escape(platnuku)))
    )

    if condition.any():
        results.append("Є власників")

# Create a new DataFrame from the results
result_df = pd.DataFrame({
    'cadastr': df['cadastr'][condition],
    'owner_dzk': df['owner_dzk'][condition],
    'owner_rrp': df['owner_rrp'][condition],
    'user_dzk': df['user_dzk'][condition],
    'user_rrp': df['user_rrp'][condition]
})

# Drop duplicates from the result DataFrame
result_df = result_df.drop_duplicates()

# Output the result DataFrame
```

Рис. 2. Приклад формування скрипта

Після чого їх потрібно загрузити у сховище даних, у даному способі використовується сховище даних Google BigQuery[1], є безкоштовні варіанти та платні його використання. Після того як дані були завантажені у сховище даних з ієрархією чи без, потрібно використовуючи SQL(Structured Query Language) впорядкувати та перетворити ці дані вже для кінцевого використання в інформаційній панелі (див. рисунок 3).

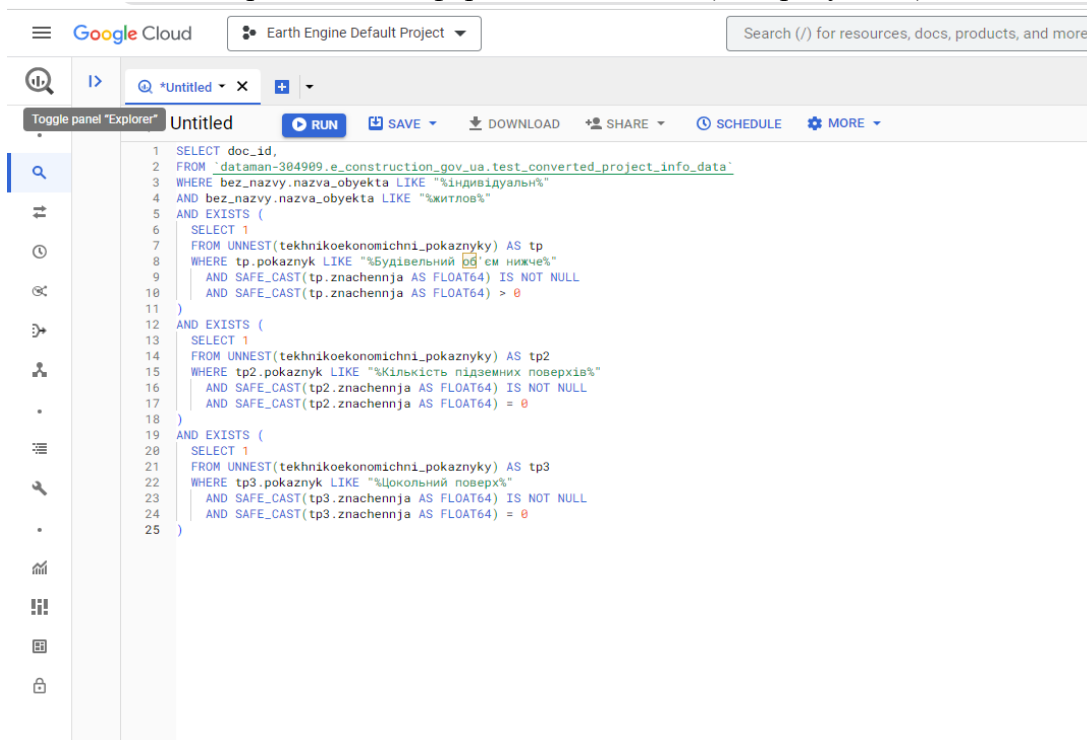


Рис. 3. Google BigQuery та SQL вибірка

Ще одним продуктом від Google для візуального представлення є Looker Studio (див. рисунок 4) [2]. За допомогою цього можна візуалізувати дані у будь-якій формі, будь то діаграма чи графік, гистограма чи таблиця і таких способів є безліч. Перевагами цієї платформи є те що вона є безкоштовною та будь-який користувач може зайти та авторизовавшись починати створювати звіти маючи хоч якусь інформацію. Недоліками є те що цей сервіс не дає можливість формувати звіти для мобільних пристроїв, існує така можливість щоб переформатувати звіт для екранів телефонів але потрібно створювати вже нову інформаційну панель та візуалізувати всю інформацію для телефонів.

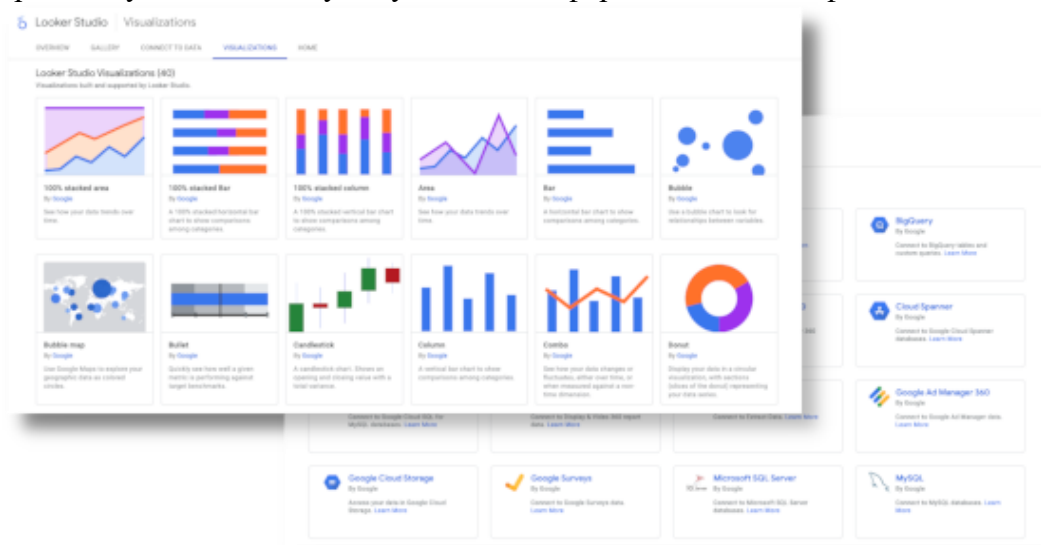


Рис. 4. Приклад звіту Looker Studio

Також є ще одна платформа для візуалізації даних це Microsoft Power BI (див. рисунок 5)[3]. Ця платформа має інший функціонал та з переваг є те що створення інформаційних панелей відбувається локально на комп'ютері без використання інтернету, якщо застосовувати бази даних які також знаходяться на комп'ютері. Також Power BI підтримує формування дашбордів для мобільних пристроїв та має застосунок для телефонів що дозволяє переглядати інформацію у будь-якому місці тримаючи у руках телефон.

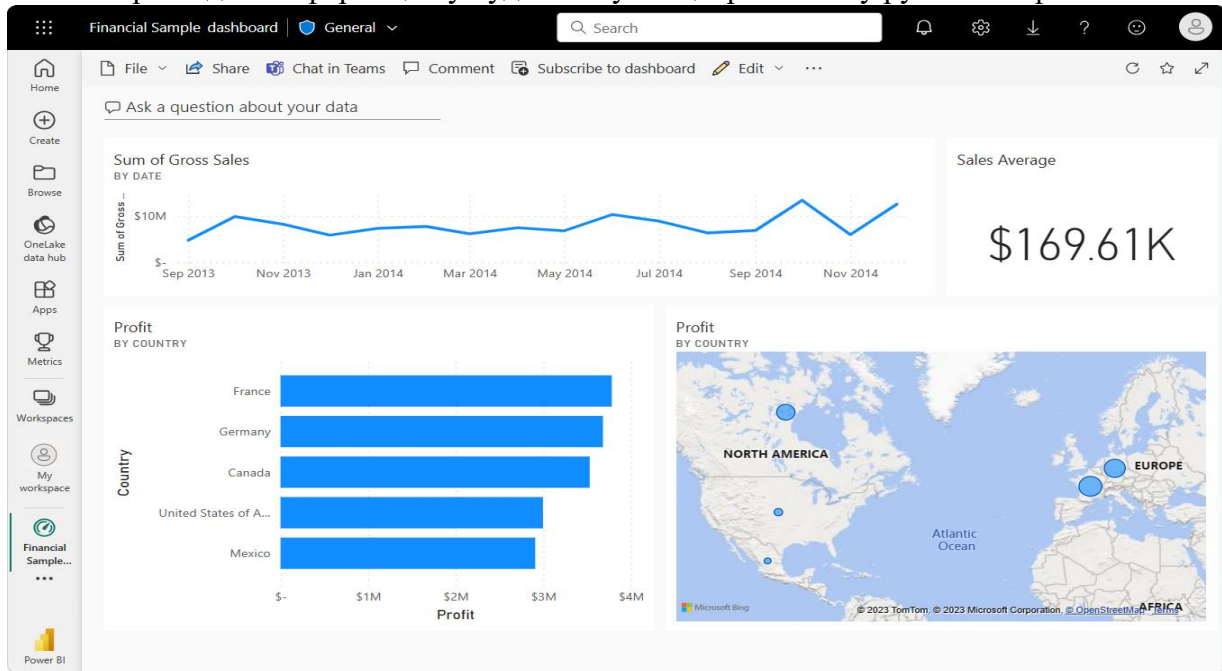


Рис. 5. Приклад звіту Power BI

Недоліком є те що це платна платформа і лише 60 днів можна отримати безкоштовний доступ, ще однією особливістю є те що вона повільніша ніж Looker Studio.

Узагальнюючи все, важливо відзначити, що в сучасному світі дані можуть мати різний вигляд і потребують впорядкування для подальшого аналізу. Використання візуальних інформаційних панелей, таких як Looker Studio чи Microsoft Power BI, є ефективним способом візуалізації даних. Обрання платформи залежить від конкретних потреб користувача, оскільки обидві мають свої переваги і недоліки. Looker Studio надає безкоштовний доступ і простоту використання, але не підтримує формування звітів для мобільних пристроїв. З іншого боку, Microsoft Power BI має більший функціонал, включаючи локальне створення інформаційних панелей і підтримку для мобільних пристроїв, але є платною платформою. У кінці року, на мою власну радість, я відзначаю свій перший рік існування та надання вам інформаційної підтримки.

Література

1. BigQuery Enterprise Data Warehouse. Google Cloud. URL: <https://cloud.google.com/bigquery>
2. Як візуалізувати дані Analytics за допомогою Looker Studio. <https://support.google.com/analytics/answer/9849873?hl=uk>. URL: <https://support.google.com/analytics/answer/9849873?hl=uk>.
3. Uncover powerful insights and turn them into impact. <https://www.microsoft.com/en-us/power-platform/products/power-bi>. URL: <https://www.microsoft.com/en-us/power-platform/products/power-bi>.