Практична робота №1

Робота з бібліотеками NumPy та Pandas

Мета роботи: отримати навички високорівневої обробки даних, фільтрації та статистичного аналізу з використанням бібліотек NumPy та Pandas на прикладі реального набору даних CDC.

Стек технологій:

Python / Pandas *для обробки структурованих даних.*

Зміст роботи

Завдання 1. Попередня обробка датасету 500_Cities_CDC.csv

Датасет містить показники здоров'я населення у 500 найбільших містах США. Кожен рядок - це унікальна комбінація міста та штату з відповідними метриками (наприклад, доступ до медицини - ACCESS2, або рівень захворюваності на артрит – ARTHRITIS). Датасет можна скачати за посиланням: https://www.kaggle.com/datasets/cdc/500-cities/data.

Великі датасети часто потребують оптимізації типів даних для економії пам'яті.

1. Завантажте файл та виведіть обсяг пам'яті, який він займає (*info*()).
2. Виявіть пропущені значення у колонках.
3. Виведіть назви міст, у яких відсутні дані одночасно і за показником *ACCESS2*, і за *ARTHRITIS*. Запропонуйте обґрунтування: чи можна видаляти такі рядки.
4. Замініть відсутні значення середнім показником по відповідному штату (використовуйте groupby та transform).
5. Розрахуйте та виведіть у відсотках, наскільки зменшився обсяг пам'яті після конвертації *Population2010* у *int32* (якщо це можливо)та текстових полів у тип *category*.

Завдання 2. Групування та векторні обчислення

Використовуючи можливості ***NumPy***, розрахуйте нову метрику.

1. Створіть нову колонку Uninsured_Population_Count, яка розраховується як:

$$Population2010 * \left(\frac{ACCESS2\_Crude\,\mathrm{Pr}\,ev}{100} \right)$$

2. Використовуючи *np.select*, створіть категоріальну ознаку City_Size:
   - Small (населення < 100,000)
   - Medium (100,000 - 500,000)
   - Large (> 500,000)
3. Виведіть топ-5 міст з найбільшою кількістю незастрахованого населення.
4. Створіть бінарну ознаку *Critical_Access*: 1, якщо рівень відсутності страхування вищий за 35%, і 0 - якщо нижчий.

Завдання 3. Статистичний аналіз за допомогою Pandas

Проаналізуйте взаємозв'язок між доступом до медицини та хронічними захворюваннями.

1. Згрупуйте дані за *StateAbbr* і знайдіть середнє значення та медіану для показника *ARTHRITIS_CrudePrev*.
2. Визначте штати, де середній рівень артриту перевищує 25%.
3. Побудуйте кореляційну матрицю (використовуючи *.corr*()) між населенням, відсутністю страхування (*ACCESS2*) та артритом (*ARTHRITIS*). Який зв'язок ви спостерігаєте?
4. Використовуючи метод Z-оцінки (або інший підхід), знайдіть міста, де показник ACCESS2_CrudePrev є аномально високим для їхнього штату.

Завдання 4. Робота з ієрархічними індексами (MultiIndex)

1. Перетворіть датафрейм, встановивши *StateAbbr* та *PlaceName* як мультиіндекс.
2. Відсортуйте дані за цими індексами.
3. Виберіть усі міста штату "CA" (Каліфорнія), де показник *ACCESS2_CrudePrev* вищий за загальнонаціональне середнє значення.

4. Обчисліть сумарне населення (Population2010) для кожного штату, використовуючи лише згрупований об'єкт мультиіндексу.

Методичні рекомендації

Поради

– Уникайте циклів for, Pandas та NumPy оптимізовані для векторних обчислень. Використання циклів на великих датасетах (наприклад, таких як CDC_500_Cities) призведе до значного сповільнення.

– Працюючи з великими даними, завжди перевіряйте типи (df.dtypes). Наприклад, заміна float64 на float32 може зменшити обсяг займаної пам'яті вдвічі без втрати точності для бізнес-аналітики.

Етап підготовки даних — фундамент успішної моделі.

Метод .info() покаже рядок *memory usage*. Щоб конвертувати типи, використовуйте *.astype*('int32').

Пряме заповнення середнім по всьому датасету є помилкою. Використання *groupby('StateAbbr')['Column'].transform('mean')* дозволяє врахувати регіональну специфіку (наприклад, рівень життя в Каліфорнії відрізняється від Міссісіпі).

Для генерації нових ознак використовуйте стандартні арифметичні операції Pandas над колонками.

Статистичний аналіз

Використовуйте *.corr(method='pearson')* для визначення кореляції. Зверніть увагу на знак коефіцієнта, якщо r > 0.7, зв'язок сильний. Наприклад, ви можете помітити, , що низький доступ до медицини (ACCESS2) корелює з вищим рівнем захворюваності.

Робота з MultiIndex

MultiIndex у Pandas дозволяють працювати з даними як з багатовимірними кубами, подібно до OLAP- cube:

- вісь 1 — штат (StateAbbr);
- вісь 2 — місто (PlaceName);
- значення — медичні показники (ACCESS2, ARTHRITIS, Population)

Створення мультиіндексу:

df.set_index(['StateAbbr', 'PlaceName']). .sort_index()

Індекс буде виглядати наступним чином:

StateAbbr PlaceName

CA Los Angeles

San Diego

San Francisco

TX Austin

NY New York

Кожен рядок → *унікальне місто в конкретному штаті.*

Далі можна «різати» куб по штату, по місту або по обох змінних одночасно (рис.1).

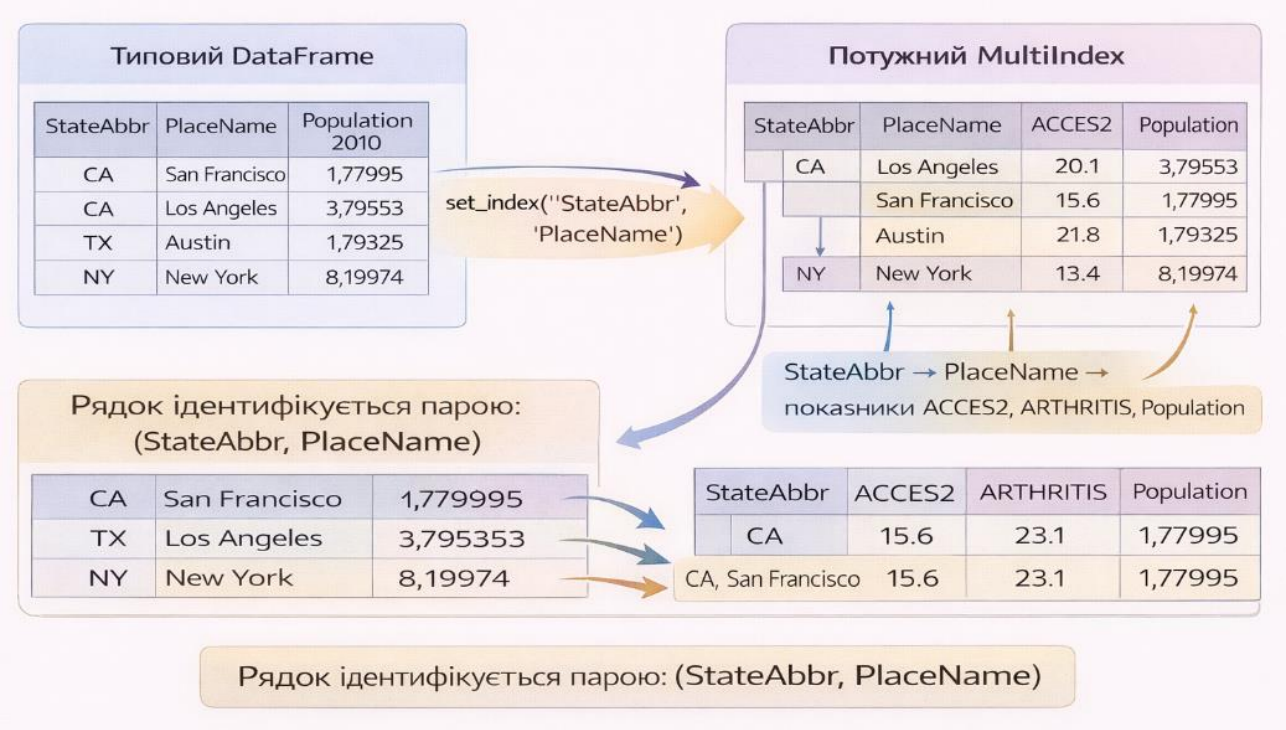

Рисунок 1.- MultiIndex у Pandas для дата сету CDC

Для вибору даних по мультиіндексу використовуйте df.xs. Це значно пришвидшує пошук у порівнянні зі звичайною фільтрацією.

Наприклад:

Встановлення мультиіндексу

```
df_indexed = df.set_index(['StateAbbr', 'PlaceName']).sort_index()
```

Сумарне населення штатів через MultiIndex

```
state_pop = df_indexed.groupby(level='StateAbbr')['Population2010'].sum()
```

Результат

```
Загальне населення по штатах (перші 5):
 StateAbbr
AK      291826
AL      965304
AR      490373
AZ     3896074
CA    22367518
```

Контрольні питання

1. Чим відрізняються методи groupby().transform() та groupby().agg()?
2. У яких випадках доцільно використовувати тип даних category у Pandas?
3. Чому np.select() ефективніший за вкладені умовні конструкції if-else?
4. Яка різниця між df.mean() та df.describe() з точки зору статистичного аналізу?
5. Що таке broadcasting у NumPy і як він використовується у Pandas?
6. Чим відрізняється df.corr() від np.corrcoef()?
7. Які ризики виникають при некоректному заповненні пропущених значень?
8. Як MultiIndex впливає на продуктивність доступу до даних?
9. У яких випадках варто використовувати xs() замість фільтрації через loc?
10. Чому середнє значення не завжди є кращою характеристикою розподілу, ніж медіана?