Лабораторна робота №4
Структури для роботи з великими обсягами даних в Python

Мета: отримати навички роботи із структурами для зберігання в Python (`python`, `numpy`, `pandas, numpy array, dataframe, timeit`)

Основні поняття: `numpy` масиви, кортежі, списки, фрейми, профілювання.

Теоретичні відомості

Мінімально необхідні навички роботи із масивами та фреймами вже отримано при виконанні лабораторних робіт 1 та 2.

За потреби можна скористатись офіційними сторінками відповідних проектів:

- http://pandas.pydata.org/pandas-docs/version/0.15.2/index.html
- https://docs.scipy.org/doc/scipy/

Хід виконання роботи

В ході виконання лабораторної роботи необхідно оцінити час виконання поставленого завдання із використанням масивів (`numpy array`) та фреймів (`pandas dataframe`).

Для кожної із структур даних потрібно виконати профілювання часу виконання (використайте `timeit` із однойменного модуля).

Дана лабораторна робота містить два рівні завдань, перший простий, другий ускладнений. Вам необхідно вибрати одне з двох: виконання першого рівня (половина оцінки) або виконання обох (максимальний бал).

Перший рівень (спрощені завдання)

Наука про дані (***Data science***) дозволяє вирішувати широке коло задач, до яких належать не тільки суто наукові задачі теоретичного плану, але й цілком практичні, які є частиною нашого звичайного життя. Оскільки наразі доволі важливою є проблема економії енергоресурсів, то розглянемо задачу, пов'язану із споживанням електричної енергії домогосподарством.

Вашій увазі пропонується набір даних ***Individual household electric power consumption Data Set***, який можна завантажити із UCI-репозиторію (за посиланням **Data folder**):
https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

Детальний опис набору даних можна знайти за наведеним вище посиланням, якщо ж "*коротко і по суті*", то це відомості щодо основних витрат електричної енергії домогосподарствами, зібрані впродовж 47 місяців (12.2006 – 11.2010).

Перелік атрибутивної інформації:

- **date**: дата виміру у форматі dd/mm/yyyy
- **time**: час у форматі hh:mm:ss
- **global_active_power**: активна потужність, яку споживає домогосподарство за хвилину (усереднено) [кВт]
- **global_reactive_power**: реактивна потужність, яку споживає домогосподарство за хвилину (усереднено) [кВт]
- **voltage**: напруга, усереднена за хвилину спостереження [В]
- **global_intensity**: усереднена силу струму для домогосподарства [А]
- **sub_metering_1**: набір споживачів енергії №1 [Вт-годин активної енергії], відповідає кухні, на якій є машина для миття посуду на мікрохвильовка (електричної плити немає, використовується газова).
- **sub_metering_2**: набір споживачів енергії №2 [Вт-годин активної енергії], відповіє пральні, в якій працює пральна машина, сушарка, холодильних та ввімкнено світло.
- **sub_metering_3**: набір споживачів енергії №3 [Вт-годин активної енергії], відповідає бойлеру та кондиціонеру.

Завдання першого рівня

Виконати всі завдання, використовуючи як `numpy array`, так і `dataframe`, проаналізувати часові витрати на виконання процедур (профілювання часу виконання), зробити висновки щодо ситуацій, в яких має сенс віддати перевагу тій чи іншій структурі даних. Висновки оформити звітом із зазначеним часом виконання та оцінкою по 5-бальній шкалі зручності виконання операцій відбору).

Також варто звернути увагу на те, що дані, як і практично все в реальному житті, можуть потребувати Вашої уваги ☺ - потрібно залишити лише ті

спостереження, в яких немає порожніх спостережень (порожні значення – пусті поля між роздільником – `?` – 28.04.2007, як приклад).

1. Обрати всі домогосподарства, у яких загальна активна споживана потужність перевищує 5 кВт.
2. Обрати всі домогосподарства, у яких вольтаж перевищую 235 В.
3. Обрати всі домогосподарства, у яких сила струму лежить в межах 19-20 А, для них виявити ті, у яких пральна машина та холодильних споживають більше, ніж бойлер та кондиціонер.
4. Обрати випадковим чином 500000 домогосподарств (без повторів елементів вибірки), для них обчислити середні величини усіх 3-х груп споживання електричної енергії, а також
5. Обрати ті домогосподарства, які після 18-00 споживають понад 6 кВт за хвилину в середньому, серед відібраних визначити ті, у яких основне споживання електроенергії у вказаний проміжок часу припадає на пральну машину, сушарку, холодильник та освітлення (група 2 є найбільшою), а потім обрати кожен третій результат із першої половини та кожен четвертий результат із другої половини.

Другий рівень (ускладнений)

Професійний Data scientist має вміти працювати з різнотипними датасетами, обробляти їх, візуалізувати та знаходити закономірності. Тому першим кроком є вибір датасету. Пропонуємо для подальших робіт вибрати датасет, що вам сподобається, із архіву https://archive.ics.uci.edu/ml/index.php/.

Датасет має відповідати таким вимогам:

- Data Set Characteristics: Multivariate
- Attribute Characteristics: Categorical, Integer, Real
- Number of Attributes: at least 2 integers/real
- Missing Values? YES!!!!!

Завдання другого рівня

Виконати всі завдання, використовуючи як `numpy array`, так і `dataframe`

1. Поборотися із зниклими даними. Для цього подивитись https://www.analyticsvidhya.com/blog/2021/05/dealing-with-missing-values-in-python-a-complete-guide/
2. Пронормувати вибраний датасет або стандартизувати його (нормалізація і стандартизація мають бути реалізовані як окремі

функції без застосування додаткових бібліотек, як наприклад sklearn.preprocessing).

3. Збудувати гістограму по одному із атрибутів, що буде показувати на кількість елементів, що знаходяться у 10 діапазонах, які ви задасте.
4. Збудувати графік залежності одного integer/real атрибута від іншого.
5. Підрахувати коефіцієнт Пірсона та Спірмена для двох integer/real атрибутів.
6. Провести One Hot Encoding категоріального string атрибуту.
7. Провести візуалізацію багатовимірних даних, використовуючи приклади, наведені у медіумі - https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57.

Додаткове завдання:

8. Поділити випадковим чином датасет на дві рівні частини. Навчити 3 регресійні моделі на основі не менше одного атрибуту відновлювати інший. Навчання має відбуватися на основі першого датасету (https://scikit-learn.org/stable/modules/linear_model.html), візуалізувати моделі та на основі середньої квадратичної помилки (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html) вибрати найкращу.