

## Хід виконання роботи ( 1 рівень )

1. Для завдання 1-3 використовуйте вибір за кондішеном. Наприклад обрати ті елементи датафрейму `df`, для яких пральна машина та холодильних споживають більше, ніж бойлер та кондиціонер:  
`df[df['sub_metering_2']>df['sub_metering_3']]`
2. Для вибору 5000 випадкових індексів використовуйте `numpy.random.choice()` з параметром `replace=True` для вибору без повторів.
3. Для вибору колонок за індексами використовуйте `df.loc()` та `df.iloc()`
4. Ускладнюйте ваші умови вибору за допомогою логічних операторів `and` та `or`

## Хід виконання роботи ( 2 рівень )

1. У вашому наборі даних обов'язково наявні пропущені дані. Вам необхідно поборотися з ними двома способами - прибрати їх, або замінити на інше значення. Якщо з першим все досить просто, то другий спосіб можна реалізувати багатьма способами : заміна пропущених значень на останнє спостереження перед ними, заміна на статистичні показники (`mean`, `median`, `mode`), або спрогнозувати значення за допомогою інтерполяції (детальніше <https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4> )
2. Провести нормалізацію або стандартизацію. Для цього використовуйте статистичні показники набору даних (`min`, `max` для нормалізації та `mean`, `std` для нормалізації):  
$$norm\_data = (data - min(data)) / (max(data) - min(data))$$
$$standart\_data = (data - mean(data)) / std(data)$$
3. У завданнях 3-4 використайте бібліотеку `seaborn` або `matplotlib` ля побудови гістограми або звичайного лінійного графіку
4. Використати `scipy.stats.pearsonr()` та `scipy.stats.spearmanr()` для обчислення коефіцієнтів Пірсона та Спірмена
5. Використати `sklearn.preprocessing.OneHotEncoder()` для проведення One Hot Encoding на категоріальних ознаках
6. Використовуйте `sns.pairplot` для побудови графіків для всіх ознак набору даних ( <https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57> )

## Додаткове завдання

1. За допомогою `sklearn.model_selection.train_test_split()` розділіть набір даних на дві рівні частини для навчання та тестування

2. За допомогою бібліотеки scikit-learn ініціалізуйте та навчіть 3 регресійні функції за допомогою першого набору даних ( [https://scikit-learn.org/stable/modules/linear\\_model.html](https://scikit-learn.org/stable/modules/linear_model.html) )
3. Візуалізуйте вашу регресійну модель ( <https://scikit-learn.org/stable/visualizations.html> )
4. Підрахуйте Mean Squared Error за допомогою **`sklearn.metrics.mean_squared_error()`** для трьох моделей за тестовими даними та виберіть найкращу модель