

У файлі `feature_selection.py`, що додається до поточного звіту, реалізовано наївний варіант алгоритму "wrapper", та застосовано його для відбору ознак у синтетичному датасеті. Для генерації даних використано функцію `sklearn.datasets.make_regression` з загальною кількістю ознак 30, з котрих значущими є 5.

На першому кроці тренуємо модель на кожній ознаці окремо і обчислюємо оцінку точності R^2 ; заносимо ці оцінки у масив `scores1`, після чого можемо визначити індекс $f1$ ознаки, для якої наближення було найбільш вдалим. На другому кроці масив значень відгука не змінюється, а модель тренуємо на двох ознаках – $f1$ та кожній із залишившихся, після чого визначаємо номер $f2$ ознаки: такої, що модель, натренована на $f1, f2$ мала найвищий R^2 . При цьому у масиві оцінок `scores2` елемент з номером $f1$ замінюємо нулем(line48) для коректного визначення індекса другої ознаки(щоб уникнути можливого здвигу). Аналогічні операції для 3-х, 4-х та 5-ти ознак дозволяють повністю позбутися незначущої інформації у згенерованих даних – достатньо перевірити `f1_score`, тобто оцінку точності моделі, натренованої тільки на відібраних ознаках, а також порівняти віднайдені індекси $f1, f2, f3, f4, f5$ з результатом стандартної функції `SelectFromModel`(окремий приклад представлено нижче).

11	0.00074983
12	2.22475e-06
13	0.00229472
14	0.138745
15	0.00383748
16	0.000301858
17	0.00172886
18	0.00308114
19	0.167754
20	0.444647
21	0.129888
22	0.00354229
23	0.174383
24	1.76849e-05
25	0.00100264

```
In [34]: featureNames.get_feature_names_out()  
Out[34]: array(['x14', 'x19', 'x20', 'x21', 'x23'], dtype=object)
```