Высокопроизводительные компьютерные технологии

Задача 1. Gzip classification

Суть алгоритма

- Между элементами тренировочного и тестового датасетов считается нормализованная дистанция сжатия **NCD**

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Здесь x, y – некоторые последовательности , xy - конкатенация,
$C(\cdot)$ – это длина последовательности сжатой с ***gzip***

- Затем применяется метод k-ближайших соседей для определения классов тестового датасета
- Метод работает на любых типах данных

Код из оригинальной статьи

```
import gzip
import numpy as np
for (x1, _) in test_set:
  Cx1 = len(gzip.compress(x1.encode()))
  distance_from_x1 = []
  for (x2, _) in training_set:
    Cx2 = len(gzip.compress(x2.encode())
    x1x2 = " ".join([x1, x2])
    Cx1x2 = len(gzip.compress(x1x2.
    encode())
    ncd = (Cx1x2 - min(Cx1,Cx2)) / max(
    Cx1, Cx2)
    distance_from_x1.append(ncd)
  sorted_idx = np.argsort(np.array(
    distance_from_x1))
  top_k_class = training_set[sorted_idx
    [:k], 1]
  predict_class = max(set(top_k_class),
    key=top_k_class.count)
```

- Красивая, но страшно медленная реализация
- Статью можно найти здесь: https://aclanthology.org/2023.findings-acl.426/

Задача

- Пользуясь numpy и joblib ускорить реализацию данного алгоритма;
- Numpy позволяет ускорить однопоточную реализацию;
- Joblib позволяет вам эту реализацию распараллелить.

Данные: последовательности ΔRR-интервалов извлеченные из ЭКГ и классы ритмов (даны в файлах *X_train, X_test, Y_train, Y_test*)
(1 – фибрилляция предсердий, 0 – нормальный ритм)

Критерии оценивания (максимум 25 баллов)

- **5 баллов** – ускорение с numpy сериальной (однопоточной) реализации
  (Обратить внимание на типы данных? Может, какие-то из операций можно заменить векторными?)
- **9 баллов** – параллелизация с joblib
  (На какие невзаимосвязанные таски можно разбить задачу?)
- **11 баллов** – отчёт о полученной производительности
  (Сравнение скорости оригинальной реализации со скоростью вашей новой сериальной реализации, ускорение на количестве ядер, графики, разница на тредах и процессах)

Пример графиков параллелизации