Duże zadanie zaliczeniowe 2. Termin 2024.06.11

W wielu zastosowaniach chcemy wykrywać zmiany zachodzących trendów. Przykładowo możemy interesować się zmianą zwyczai wydatkowych użytkowników karty kredytowych z całego świata. Jeżeli obserwujemy strumienie danych (np. z kwotami konkretnych płatności) to możemy użyć w tym celu Biased Reservoir Sampling (BRS).

Do zadania jest załączony skrypt generujący dwa strumienie danych w postaci (timepoint : int, value: float). Dane są posortowane, ale może brakować danych dla niektórych punktów czasu. Twoim zadaniem jest zaimplementowanie w sparku algorytmu próbkującego niezależnie każdy ze strumieni z limitem rozmiaru próbki N=1000 i z użyciem algorytmu opisanego w https://charuaggarwal.net/sigreservoir.pdf. Równolegle należy próbkować dane używając zwykłego Reservoir Sampling (RS).

Co 100k jednostek czasu (rozpoznajemy to, jak przyjdzie pierwszy element z wartością timepoint >100k, >200k etc) dla każdego strumienia i kazdej metody próbkowania należy wykonać test zgodności rozkładów Kołmogorova-Smirnova (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html) na zgodność rozkładu próbek z obu strumieni oraz z zapisanymi próbkami z maksymalnie 4 wcześniejszych momentów pomiarowych dla każdego ze strumieni (maksymalnie 18 wykonań testu KS dla danego momentu czasu). Celem jest sprawdzenie, czy strumienie zmieniają swój rozkład i czy dwa strumienie są próbkowane ze wspólnego rozkładu. Próbkowanie należy zakończyć po 20-krotnym wykonaniu porównania (powyżej momentu 2 000 000)

Dane z generatora mają być przekazywane do analizy za pomocą strumieni Kafki, z której dane mają być czytane i analizowane za pomocą sparka. Zadbaj o to, aby twoje rozwiązania było skalowalne na wiele maszyn.

W folderze znajduje się skrypt `data_generator.py`, który ma posłużyć jako źródło danych.