**GROUP NUMBER:** 97

**GROUP MEMBERS:** Bianchin Umberto, Vettorello Massimo

**AVAILABLE INPUTS:** Input files are available in the hdfs file system: /data/BDC2425/artificial1M7D100K.txt and /data/BDC2425/artificial4M7D100K.txt

**PART 1:** The goal of this test is to assess the scalability of the standard and fair implementations. The test must be performed on file artificial4M7D100K.txt. However, if your implementation is slow (i.e., taking more than 10 minutes for the slowest run), you can use the smaller file artificial1M7D100K.txt. You must use the following parameters: L=16, K=100, M=10.

Fill in the following table.

**Name of used file:** artificial1M7D100K.txt

|  |  |  |  |
| --- | --- | --- | --- |
| **SCALABILITY WITH RESPECT TO NUMBER OF EXECUTORS** | | | |
| **Number of executors** | **Spark Lloyd’s implementation** | **MRFairLloyd** | **MRComputeFairObjective** |
| 2 | 11202 ms | 1526383 ms | 166608 ms |
| 4 | 8154 ms | 806848 ms | 87005 ms |
| 8 | 4245 ms | 398073 ms | 42311 ms |
| 16 | 2867 ms | 201968 ms | 21490 ms |

**General hints:**

* Remember that Spark uses the lazy evaluation for constructing an RDD. Therefore, be sure to include an action on the final RDD when you take running times.
* Any used RDD in your program should be cached.
* Do not include the reading of the input in your running times.

**PART 2:** Describe the program GxxGEN.java or GxxGEN.py that you have implemented for point 5 of the specifications. Include a brief high-level description of your program and the constraints, if any, on the input parameters (e.g., the minimum number of points *N*).

The program G97GEN.py is a synthetic data generator designed to create a 2D dataset for evaluating fairness in clustering algorithms, specifically comparing standard Lloyd's algorithm and its fair variant. This dataset intentionally includes demographic imbalance and geometric structure to highlight the strengths and limitations of fair clustering techniques.

**High-Level description**

The script generates a total of N points in two-dimensional space, divided into:

* **Group A (majority group):** formed by K-1 compact clusters evenly distributed in a circular pattern.
* **Group B (minority group):** formed by a single, dispersed cluster centered at the origin.

Each point is labeled either “A” or “B” to denote its demographic group, the final dataset is saved as a CSV file (datasets/G97GEN.csv) with each row representing a point in the format (x, y, label).

This structure is useful for illustrating disparities in clustering performance between algorithms that do and do not account for fairness.

**Input parameters:**

This program takes two command-line arguments:

1. N = Total number of points to generate, must be a positive integer
2. K = Total number of clusters, must be an integer greater than 1 (at least one group is reserved for group B and the rest for group A)

**Additional Characteristics:**

* A fixed minority fraction β = 0.1 means approximately 10% of the points belong to group B.
* Group A clusters are placed on a circle of radius R = 50.0 and use a standard deviation σ\_A = 1.0.
* Group B points are sampled from a wider Gaussian with σ\_B = 15.0 centered at the origin.
* The dataset is randomly shuffled before being printed and saved.