MIE1628

Due Date: June 5, 2025

Assignment 1

Clustering Techniques with Hadoop MapReduce (70 marks)

- **No** submission is accepted via email. We have **no exception**.
- Students are responsible for submitting the **correct files** **on time**.
- Assignments submitted up to 48 hours late will incur a **20% penalty**.
- Your grade will be **zero** if you submit your answer **after 48 hours**.

Contact your TA for any questions related to this assignment or post clarification questions to the Piazza platform.

Java programming language is recommended for this assignment, but you can use python as well. This assignment explores the application of k-means clustering and canopy selection within the MapReduce framework. It emphasizes a deeper understanding of the algorithms, their limitations, and efficient implementation strategies. All code should be well-documented and follow best practices. You must clearly explain your design choices and justify your implementation decisions in your report.

Part 1: Line Counting with MapReduce (20 marks)

1. **(15 marks)** Implement a MapReduce program to count the number of lines in a large text file (**shakespeare.txt**). Analyze the performance of your implementation, considering factors such as the number of mappers and reducers, input file size, and network communication overhead. Provide a detailed performance analysis, including graphs as needed and a discussion of optimization strategies. (Implement using Hadoop MapReduce)

2. **(5 marks)** Propose at least one optimization strategy to improve the efficiency of your line counting MapReduce program. Justify your choice of optimization and quantify its impact on performance. (Describe in words)

Part 2: K-Means Clustering on MapReduce (30 marks)

3. **(5 marks)** Propose a distributed k-means clustering algorithm using MapReduce. Use the provided dataset (**data_points.txt**). Your implementation should handle a variable number of clusters. Thoroughly explain your algorithm, including the partitioning strategy, centroid calculation, and convergence criteria. Discuss the choice of distance metric and its rationale. (Describe in words)

4. **(20 marks)** Experiment with different values of k = 5 and 9). For each k, report the cluster centroids, the number of iterations required for convergence (or the maximum

iterations reached), the computation time, and a qualitative analysis of the resulting clusters. Visualize your results where possible (e.g., scatter plot of data points with cluster assignments). Analyze the impact of k on the quality of the clustering results and the computational cost. (Implement using Hadoop MapReduce)

5. **(5 marks)** Critically evaluate the performance of your k-means implementation. Discuss the impact of data distribution and the choice of distance metric (Euclidean, Manhattan, etc.) on the algorithm's performance and convergence. Analyze the scalability of your implementation – how does runtime change if you increase the dataset size? (Describe in words)

Part 3: Canopy Clustering and Optimization (15 marks)
Read the provided paper, research as needed and then answer the below questions in words.

6. **(5 marks)** Explain the advantages and disadvantages of using k-means clustering with MapReduce. Discuss the trade-offs between parallelization, communication overhead, and the inherent limitations of the k-means algorithm itself. (Describe in words)

7. **(5 marks)** How do you implement Canopy Clustering as a pre-processing step for k-means. Justify your choice of distance metrics for the canopy and k-means stages. Explain how your implementation reduces the number of distance comparisons in the subsequent k-means phase. Clearly explain the parameters used for Canopy Clustering and their impact on the results. (Describe in words)

8. **(5 marks)** How do you integrate Canopy Clustering into your MapReduce-based k-means algorithm. Compare the performance (runtime and cluster quality) of k-means with and without Canopy Clustering as a pre-processing step. (Describe in words)

Deliverables:

- **(5 marks)** A detailed report explaining your approach, methodology, results, analysis, and conclusions. Include visualizations, tables, graphs, and performance measurements as appropriate.
    - Your report should demonstrate a clear understanding of the algorithms, their limitations, and the practical challenges of implementing them in a distributed environment.
    - A code file (well-commented and organized) should be submitted along with the runtime output screenshots.
    - Include references as appropriate.