

Final Project Report

Note: You can access the code via [this GitHub repository](#).

Introduction:

The software system described herein serves as a robust platform for conducting clustering and heuristic operations through an intuitive user interface. By integrating various algorithms and data manipulation functionalities, it empowers users to explore and analyze datasets effectively. This report delves into the system's objectives, features, and functionalities while omitting detailed class designs and specific results.

Objective:

The primary objective of this software system is to provide a user-friendly environment for performing clustering and heuristic operations on datasets. By encapsulating complex algorithms and data processing tasks into intuitive interfaces, the system aims to facilitate data analysis, exploration, and decision-making processes. Through efficient implementation and abstraction of algorithms, it aims to offer scalability and adaptability to diverse datasets and analytical scenarios.

Class Functionalities

User Interface (UI) Class:

- Initializes the UI with a template path.
- Manages UI elements such as buttons, plotting, and data panels.
- Handles user interactions like manual run, clustering, heuristics, etc.
- Provides methods to update UI components based on operations.
- Handles loading and displaying data.
- Manages the main window and its functionalities.

Point Matrix Class:

- Loads, saves, and clears data.
- Manages data related to points and clusters.
- Calculates distances, centers, and objectives.
- Handles data manipulation and storage.

Point Class:

- Represents individual points with coordinates and cluster ID.
- Provides methods for setting and getting cluster IDs and coordinates.

Heuristic Operations Class:

- Inherits from Point Matrix and handles heuristic algorithms.
- Implements methods like hill climbing, simulated annealing, etc.
- Calculates objective functions and optimal clusters.
- Manages cluster hubs and node swapping/relocation.

Clustering Operations Class:

- Inherits from Point Matrix and deals with clustering algorithms.
- Implements clustering methods like k-means, affinity propagation, etc.
- Configures algorithm parameters and performs clustering.
- Handles different clustering techniques and their settings.

Class Hierarchy:

1. UI_Interface:

- Inherits from QMainWindow, Clustering_Operations, and Heuristic_Operations.
- UI_Interface integrates functionalities from both clustering and heuristic operations.
- Manages the main application window and user interactions.
- Calls methods from Clustering_Operations and Heuristic_Operations to perform clustering and heuristic tasks based on user input.

2. Point_Matrix:

- Acts as a core data structure for managing points and clusters.
- Provides foundational methods for data loading, saving, and processing.
- Clustering_Operations and Heuristic_Operations classes inherit from Point_Matrix.

3. Point:

- Represents individual data points within the Point_Matrix.
- Each point has coordinates and a cluster ID, and methods to manipulate these attributes.

4. Clustering_Operations:

- Inherits from Point_Matrix.
- Implements various clustering algorithms.
- Provides methods to configure and execute clustering tasks.

5. Heuristic_Operations:

- Inherits from Point_Matrix.
- Implements heuristic algorithms for optimizing clustering results.
- Provides methods to configure and execute heuristic optimization tasks.

6. Get_Data_Dialog_Class:

- An **external** component for facilitating data input.
- Provides a dialog window for users.
- Allows users to specify function parameters.

7. Progress_Show_Function:

- An **external** component for displaying progress during lengthy operations.
- Provides a function or module to show progress bars or status indicators.
- Enhances user experience by providing feedback on the progress of operations

Essential Dependencies:

1. PyQt5:

- Used for creating the graphical user interface (GUI).
- Provides the QMainWindow class for the main application window and other UI components.

2. NumPy:

- Used for numerical operations and data manipulation.
- Provides support for arrays and mathematical functions.

3. Matplotlib:

- Used for plotting and visualizing data.
- Provides functions to create static, animated, and interactive visualizations.

4. OpenCV:

- Used for image processing tasks.
- Provides functionalities to convert images to pixmaps for display in the UI.

5. Scikit-learn:

- Used for implementing clustering algorithms.
- Provides a wide range of machine learning and clustering tools, including k-means, DBSCAN, and more.