REDUCING SEARCH SPACE IN BIG DATA

PRIYA SHARMA 1MS12CS144

SWEETY KUMARI 1MS12CS119

SATHVE V G 1MS12IS117

SURABHI KUMARI 1MS12IS110

SOFTWARE REQUIREMENT SPECIFICATION

**1. Project Overview**

The project is to reduce the huge search space to make searching easier and efficient. It can be used in Various fields to make information available readily. It has great scope in many fields and can reduce losses in everything. It helps to make the searching efficient faster and better. The main objective of this project is to mine frequent itemsets to reduce the search space so as to avoid wasting lots of time and space in computing all frequent patterns first and removing uninteresting ones as a post-processing step. This project aims to first find appropriate algorithms to implement. Then find dataset ts to implement and which can be useful and applicable to the society. Choose platforms to implement these algorithms to get proper results. Compare the various methods by drawing graphs and using various sizes of data.

**2. External Interface Requirements**

**2.1. Hardware Interfaces**:

The application is intended to be a stand-alone, single-user system. The application will run on Windows operating system. No further hardware devices or interfaces will be required.

**2.2. Software Interfaces**

**Input**

The software will receive input from two sources. First dataset which we are feeding into our matlab for performing clustering using Fuzzy-c means. Second we are feeding output of our matlab program as input into our java program for searching our keywords.

**Output**

The output will display the result of searching but with reduces time complexity. Basically we are trying to show that by first clustering dataset , searching will take less time.

**Operating System** The software will run on Windows.

**2.3. User Interfaces**

The interface will meet the following requirements to conform to the users’ needs. It will be simple and easy to understand. Controls which allow the user to interact with the application will be clear and imply their functionality within the application. The interface will include user inputs as well as graphics. The graphics displayed to the user will provide a visual representation of the output produced.

User Inputs (Mandatory)

The user change dataset to see how fast our clustering and searching algorithm works.

Graphic: Cluster Centers(Mandatory)

After performing Clustering of dataset, we are able see Centers of our Clusters. We are saving centers in database we can be used by the searching program.

**2.4. Communication Interfaces**

Our project consist of 2 interfaces. First one performs clustering and second one perform Searching. Proper communication is needed between these modules as output of First interface is feeded as input of second interface.

**3. FUNCTIONAL REQUIREMENTS**

**3.1 Pre Processing**

To convert raw data into processed data we are using Rapid Miner to eliminate noise, redundant data and to handle undefined data.

**3.2 Clustering**

After getting the processed data we form clusters of this data using FUZZY C MEANS algorithm. The FCM algorithm attempts to partition a finite collection of elements into a collection of c fuzzy clusters with respect to some given criterion.

**3.3 Searching**

A given input data element is searched by first locating the cluster it belongs to by means of Fuzzy c algorithm and then the data element is searched within that cluster by using searching algorithm. Thus reducing search space from all the clusters to a particular cluster.

**4. Software System Attributes**

**4.1. Reliability**

The software will meet all of the functional requirements without any unexpected behavior. At no time should the output display incorrect or outdated information.

**4.2. Availability**

The software will be available at all times on the OS, as long as the device is in proper working order. The functionality of the software will depend on any external services such as internet access that are required. If those services are unavailable, the user should be alerted.

**4.3 Security**

The software should never change user Dataset or produce wrong Cluster Centers.

**4.4.Maintainability**

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy.

**4.5. Portability**

This software will be designed to run on Windows OS and forward compatible.

**4.6 Performance**

Performance of our project depends upon how fast and efficient our clustering algorithm works.

**5. Performance Requirements**

In this project, performance is basically related to the searching and clustering algorithms that can be used for implementation.

There are various factors that affect the performance in this project are:-

1. **Large dataset storage**  :- Need to store and verify more data and need to do it faster.
2. **Data ingestion**  :- system should be able to consume the data faster from the data source. It also includes how quickly data can be inserted into underlying data store .
3. **Data Processing**
4. **Application of appropriate clustering and searching algorithms:-**

**Basically relates to the efficiency and complexity of the algorithms used . Clustering algorithm should be chosen such that data should be partitioned properly and so that searching becomes easy and efficient.**

**6. Database Requirements**

For this project we are not using any database but we are making use of file system for giving input dataset and storing the result of intermediate processes

**7. Design Constraints**

We are using file system for storing result of Clustering algorithm. If size of dataset is very huge then file System won’t be efficient.

**8. Other Requirements**

We have taken the dataset related to CO2 emission from manufacturing industries and its effect on environment and surroundings. For above data processing techniques to be applied on the data first cleansing of data is required and for data cleaning we apply various preprocessing techniques on our input raw-data.

As a part of data preprocessing we apply various techniques like Replacing missing value, feature extraction etc.

Applying missing value replacement on the data below:-

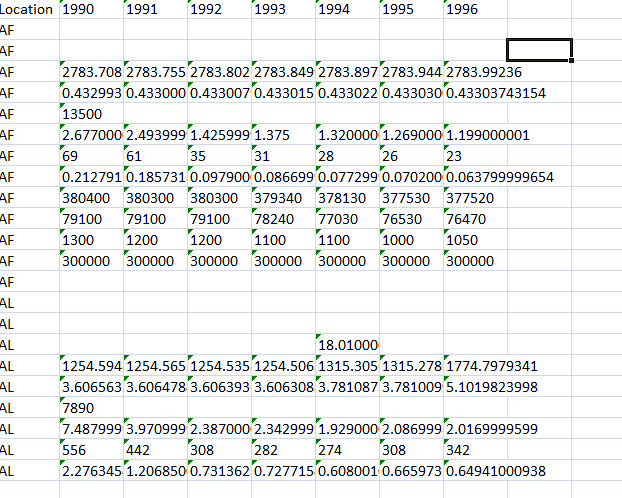


Fig 6: input dataset

And as a result of applying missing values replacement methods on the data given above,the resultant processed data we get is given below:-



Fig 7: result after preprocessing