CHAPTER - 4

CLUSTERING ALGORITHM FOR MIXED DATA

4.1 Introduction

Clustering analysis is a fundamental method in data mining. Its goal is to understand the properties of clusters of data entities within the attribute space. Clustering algorithms [148] find applications in diverse fields, involving research on social network analysis [149], uncovering knowledge, image manipulation, and textual as well as sentiment analysis [150]. The main objective of clustering analysis is to divide data objects with distinct characteristics into different clusters while grouping those with similar attributes together. The two primary clustering techniques [151] are hierarchical and partitional methods. Hierarchical clustering methods distribute data into a hierarchical diagram of stratified sections utilizing a splitting or accumulative process. On the other hand, by reducing an objective cost function, partitional clustering algorithms divide data into a specified number of groups.

Clustering algorithms are capable of handling specific types of data. Numerical data is made up of continuous values, unlike categorical data, which is composed of discrete data, which can only have a finite number of values. Categorical data, such as name, gender, and educational attainment, are used in a wide range of applications in the real world. Mixed datasets have both categorical and numerical values. Real-world data frequently comes in a variety of forms. For instance, categorical and numerical variables in medical data include age, height, weight, salary, nationality, gender, employment, education [152], marital status, and kind of chest pain [153]. Comparing two sets of data gets more challenging when a dataset contains categories and numerical variables [154]. To address the similarity issue, divide a mixed dataset into numerical and categorical components and compute the For

numerical qualities, the Hamming distance and the Euclidean distance between two numerical and categorical data points, respectively [155].

Partitional clustering algorithms [156] for mixed-type datasets undergo repeated assignment and update phases using complete data sets until their stopping criteria are met, making them unsuitable for rapidly clustering very large datasets. The clustering time is dramatically extended by this method. A proximity matrix is necessary for hierarchical clustering techniques [155] for mixed data sets, and creating the matrix involves significant computing expenses. Moreover, a sizable amount of RAM must be set aside to keep the matrix active until the clustering is finished. Therefore, the hierarchical clustering approaches are inappropriate for grouping large mixed-type data sets.

Because of its efficacy and simplicity, k-means is a well-liked partitional clustering technique. approach in many disciplines. Huang [157] developed the well-known k-prototypes method for clustering heterogeneous data, which combines the k-means and k-modes methodologies. To improve the k-prototypes algorithm, [158] incorporated the cluster centre representation and attribute influence. Chen pioneered self-adaptive peak density clustering, and He [159] expanded on the density clustering concept. Two primary objectives are shared by many mixed data clustering strategies: In order to cluster data in a way that yields the best local result, it is first necessary to provide novel metrics of similarity among mixed features.

The Reclust approach for mixed data of numbers and categories data, which includes repeated clustering and validation of cluster, is introduced in this chapter. The method includes three critical steps:1. initial clustering,2. validation, and 3. re-clustering. The first clustering step employs two clustering algorithms: Modified Probability and Similarity-based K-Means (MPS-KM) and Modified-Self-organizing Map (MSOM). The validation process that follows assesses the clustering result. Finally, the wrongly

clustered data is re-cluster by using the re-clustering technique. Cluster outcome quality is increased by iterative validation and re-clustering processes.

4.2 Mixed Data Clustering

An unsupervised machine learning method called clustering organizes unlabeled data into groups of data points that are "similar" to one another and "dissimilar" from those in other clusters [160]. Several clustering techniques can only work with data that have either categorical or numeric attribute values. Mixed datasets are common in the real world and include categorical and numerical information. Many applications, including those in finance, marketing, and health, commonly use mixed data [161]. Thus, creating machine learning algorithms that can manage such data has become crucial. Clustering becomes more challenging when a dataset contains numerical and categorical variables. In such cases, devising innovative methods to measure similarity among mixed features and applying clustering techniques using existing or rising techniques are the primary objectives of most mixed data clustering algorithms. Some of the earliest mixed clustering systems used direct adaptations of partitional clustering algorithms [162].

In their study, Que et al. [163] present a similarity metric that employs entropy-based weighting for clustering mixed datasets. They begin by employing an automatic categorization method to change numerical data into categorical data. Then, they utilize an entropy-based weighting method to indicate the importance of different attributes.

Li et al. offer a repeated weight alteration technique that incorporates a centroid for the noise-filtered distribution. [164] as a mixed data clustering strategy. For categorical attributes, it explains the centroid of a noise-filtered distribution. This method identifies the cluster centre with mixed attributes by fusing the mean and noise-filtered distribution centroid. Additionally, the centroid of the noise-filtered distribution provides a more precise account of

the frequency of instances for each potential value of a cluster's categorical features.

Jia and Cheung's [165] example of using numerical and categorical variables with soft subspace clustering to cluster data. Based on the idea of object-cluster similarity, the model includes attribute weighting. To compute each feature's contribution to the cluster, taking into consideration both inter- and intra-cluster variation, a uniform weighting approach for numerical and categorical variables is utilised.

D'Urso and Massari [166] present a fuzzy clustering model for data with heterogeneous features that takes several variables or qualities into account throughout the clustering process. A weighting scheme is used in this model to aggregate dissimilarity measures for each feature, resulting in a distance scale for a number of attributes. The weights are calculated objectively during the optimisation process and indicate the relevance of each attribute type in the clustering results. Rodriguez et al. [167], on the other hand, provide a multipartition clustering approach that handles mixed data efficiently by integrating Bayesian network factorization with the variational Bayes framework.

Ryu et al. [168] propose an innovative summary-based clustering technique that efficiently clusters very large mixed-type datasets. The approach incorporates an adaptively expanding memory size to achieve more precise threshold estimation. It utilizes a histogram to represent the mixed-type data and a cluster feature vector.

Additionally, Ji et al. [169] introduce a novel multi-view clustering approach. The k-prototypes approach for clustering data with numerical and categorical attributes is used for the first time in a multi-view format. The author develops the strategy to acquire the final clustering result by combining the clustering results on each view and also presents a representation

prototype and updating approaches for the cluster centres under the scenario of many views.

A sample clustering algorithm that makes use of deep-learning frameworks is deep-embedded clustering. Only applicable to numerical data, it simultaneously learns low-dimensional feature representations and maximizes the clustering objectives. A new deep-embedded clustering methodology that works with both numerical and categorical data was proposed by Lee et al. [170].Furthermore, soft-target updates can use mixed data to promote convergence stability; this idea was taken from an enhanced deep Q-learning method used in reinforcement learning.

Dinh et al. propose a new strategy to group missing values in mixed numerical and category data in [153]. Their method combines the grouping and restoration phases into a single procedure, comprising three distinct steps. First, the initialization step based on the missing object and attribute type values, separates the input dataset into two pieces. The collection of connected data objects is located during the imputation phase using a decision-tree-based technique. Using mean and kernel-based techniques, cluster centres are finally generated at numerical and categorical attributes during the clustering phase.

Ji et al. [148] present a unique partitional clustering technique for mixed numerical and categorical data based on Cuckoo Search and K-Prototypes. The authors create a novel representation for candidate solutions that accommodates numerous attributes and propose two algorithms for the cuckoo to use when looking for prospective answers, either near existing solutions or throughout the entire attribute space. Furthermore, Ji et al. [171] present a novel clustering approach dubbed ABC-K-Prototypes. This method combines the k-prototypes technique, the ABC optimisation strategy, and chaos theory. To address the mixed numeric and categorical data, the one-step k-prototypes process is described first, and this process is then merged with the search tactic of the artificial bee colony. Furthermore, the scout bees'

search operation employs chaotic maps to generate chaotic sequences as replacements for random integers. Using multi-source search during the scout bees' search phase improves the algorithm's convergence even further. Table 4.1 summarises the advantages and downsides of several past works.

Table 4. 1 Pros and Cons of Mixed Data Clustering

Algorithms	Advantages	Disadvantages
CCS-K-Prototypes [148]	It reduces time and space complexity. It increases accuracy, precision, recall and randindex.	Performance is slowed down by the multi-core system's data transfer between its numerous processors and main memory.
k-CMM [153]	This approach exhibits higher efficiency compared to k-prototypes and leads to improved clustering excellence.	It cannot deal with datasets with unstructured forms and missing features.
K-mixed prototype [156]	The K-mixed prototypes improved the clustering accuracy by giving binary type traits greater weight than nominal type attributes.	Large datasets cannot be clustered effectively
Multipartition clustering [167]	Increases clustering quality	Divergence problem occurs
Summary-based clustering [168]	It decreases clustering time and considerably increases clustering accuracy.	It cannot handle a very large dataset.

4.3 Data Preprocessing

Data preprocessing is an evolutionary procedure that transforms raw data into comprehensible and valuable formats. Nonetheless, raw datasets often exhibit flaws, inconsistencies, and lack behavioral patterns and trends.

Therefore, building correct ML models requires a crucial step.It entails activities like cleaning, transformation, integration, and reduction because incompleteness, noise, ambiguity, and inconsistency might be found in a dataset. Therefore, this research considers data cleaning (missing value handling) and transformation (Numerical to categorical conversion).

The process of data cleansing involves identifying inaccurate or noisy data and correcting or eliminating it from the dataset. It focuses primarily on locating and replacing erroneous, irrelevant, or otherwise noise-filled data and records.In a dataset, it is typical for certain columns to have missing values. The problem may have been caused by data-gathering procedures or data validation processes. Nonetheless, it is important to consider missing values because they may lead to a model's feature being eliminated. Simple interpolation techniques can fill such matters if missing a respectable quantity of values. Using mean, median, or mode values for model feature values is the most popular approach for handling it.

The most challenging problem in statistical analysis and machine learning arises from missing values. There are numerous methods for handling missing values because it is crucial to find useful information. However, the most common approach to dealing with a large dataset is to delete instances with missing values. In such cases, removing instances with missing values leads to the loss of significant data, adversely impacting algorithm performance. As a result, handling the missing values requires an effective strategy.

This research work usesa simple method for handling missing data. For numerical attributes, use the mean value of a similar class for the missing attribute and for the categorical attribute, use the frequent high value of a similar class. For example, consider the sample dataset shown in Table 4.2

Table 4. 2 Sample Dataset with Missing Value

Age	Educational_Gap	Course_Choice	Family_Income	Class
18	No	Passion	Rich	High
20	No	Compulsion	Poor_class	Average
17	Yes	Passion	Below_Poverty_Line	Average
19	Yes	Passion	Middle_class	High
20	No	Passion	Poor_class	Low
21	No	Passion	Poor_class	Low
?	No	Compulsion	Below_Poverty_Line	High
18	No	?	Middle_class	High
20	?	Compulsion	Poor_class	Low

The missing value in the age attribute is imputed with the mean value of the High class. For example, $\left\lceil \frac{18+19+18}{3} \right\rceil = 19$. The missing value of the Course_Choice attribute is filled with the high frequency of class High. The Course_Choice is filled with 'Passion'. The missing value of Educational_Gap is filled with the high-class average frequency, i.e.'No' is filled with Educational_Gap. Table 4.3 is shown the dataset after imputation.

Table 4. 3 Dataset after missing value imputation

Age	Educational_Gap	Course_Choice	Family_Income	Class
18	No	Passion	Rich	High
20	No	Compulsion	Poor_class	Average
17	Yes	Passion	Below_Poverty_Line	Average
19	Yes	Passion	Middle_class	High
20	No	Passion	Poor_class	Low
21	No	Passion	Poor_class	Low
19	No	Compulsion	Below_Poverty_Line	High
18	No	Passion	Middle_class	High
20	No	Compulsion	Poor_class	Low

At the outset of the work, a crucial consideration is transforming data into suitable formats for a specific machine-learning task. The model's outcomes and data mining are significantly influenced by the inclusion of

irrelevant, redundant, noisy, and unreliable data, which complicates the training phase. In addition, the format of the data must be suitable for ML. For instance, if the algorithm analyses numerical input, a class labelled 'Low' or 'High' must be changed to "0" or "1". This research work uses discretization.

By splitting the range of a continuous property into intervals, data discretization techniques are used to reduce the number of values for the attribute. Then, instead of using the actual data values, interval labels might be used. The discretization procedure involves dividing the continuous attribute's ranges into intervals.For example, the age attribute value is {18, 20, 17, 19, 20, 21, 19, 18, 20} discretized into {17.5-19.5, 19.5-inf, inf-17.5, 17.5-19.5, 19.5-inf, 19.5-inf,17.5-19.5, 17.5-19.5, 19.5-inf}.

4.4 Proposed Methodology

This section introduces the Reclust clustering algorithm, which includes re-clustering and cluster validation, for hyrbid datanumbers and categories . The three main steps of the suggested technique are initial clustering, validation, and re-clustering. Utilising MPS-KM and MSOM, a pair of clustering techniques, is part of the initial clustering step. The validation phase then evaluates the results of the clustering, and the re-clustering procedure deals with material that was erroneously clustered. Iteratively carried out validation and re-clustering procedures significantly improve the quality of cluster results.

A dataset containing n instances with mixed data is represented as D, denoted as $\{d_1, d_2, \ldots, d_n\}$. a_c categorical attributes and a_u numerical attributes are contained in dataset D. After that $d_i(1 \leq i \leq n)$ could be indicated as $[d_i^c, d_i^u]$ with $d_i^c = \left[d_{i1}^c, d_{i2}^c, \ldots, d_{i,a_c}^c\right]$ and $d_i^u = \left[d_{i1}^u, d_{i2}^u, \ldots, d_{i,a_u}^u\right]$. The goal is to form k clusters by clustering the dataset D. C= $\{C_1, C_2, \ldots, C_k\}$. $C_i \cap C_j = \emptyset$, $\bigcup_{i=1}^{k} C_i = C(i, j = 1,2, \ldots, k, i \neq j)$.

The Reclust clustering algorithm is described in Algorithm-1.

Algorithm-1 Reclust

Input: *Dataset D = { data1, data2, …, data_n}, Cluster Count num_clusters*

Output: *Clustering Outcome*

1. Cluster initialization

    1a. KM_clusters = Apply MPS-KM(D, num_clusters)

    1b. SOM_clusters = Apply MSOM(D, num_clusters)

2. Validation of clusters

    2a. KM_eval = evaluateCluster(KM_clusters, D)

    2b. SOM_eval = evaluateCluster(SOM_clusters, D)

    2c. min_eval = Min (KM_eval, SOM_eval)

    2d. incorrect_data = incorrectlyClusteredData(D)

3. Re-clustering

    3a. While (the termination condition is not satisfied)

    3b. re_clusters = Cluster incorrect_data using min_eval

    3c. re_eval = evaluateCluster (re_clusters)

    3d. sub_data = incorrectlyClusteredData(incorrect_data)

    3e. incorrect_data = sub_data

    3f. End While

MPS-KM and MSOM are two clustering techniques used in Step 1. Step 2 evaluates the cluster outputs' quality. A class-based cluster evaluation method is used by the evaluate Cluster function. The class attribute is disregarded during the cluster formation process, and classes are then assigned to the clusters during the testing phase based on the predominance of the class feature inside each cluster. This assignment determines how to calculate the categorization error. The least error between the two clustering techniques is found in Step 2c. Step 2d uses the evaluation findings to pinpoint the data that were wrongly grouped. In Step 3, the inaccurate information is repeatedly re-clustered, and the clustering results are assessed. If the stop requirement is not

met, the re-clustering procedure proceeds. R Either a minimal error measure or a minimum amount of data points in the incorrectly grouped data must be met to satisfy the stop requirement for re-clustering to stop.

The modified K-means clustering is explained in Algorithm-2. Euclidean distance is utilized in conventional k-means clustering to calculate the distance between two instances. This work uses the probability and similarity based distance function for distance computation.

Algorithm-2MPS-KM

Input: Dataset *X = { x1, x2, …, xn}, Number of Cluster k*

Output: *Clustering Outcome*

1. Discretize numerical attributes
2. Compute the distance between two categorical values using Algorithm3
3. Randomly assign each instance to a different cluster (k)
4. Find the k cluster centre
5. For each instance in X
6. For each cluster centre
7. Find probSim=dist(d_i,C_j)+Sim(d_i+C_j)
8. EndFor
9. Assign d_i to the closest cluster centre
10. EndFor
11. Re-calculate cluster centre
12. Repeat steps 5 to 11 until the stop criterion is met

In algorithm-2, the numerical attributes are converted into Discretize attributes. Then, the distance between two categorical values is computed using algorithm-3. Initially, the k-cluster is formed randomly. Then, the instances are randomly assigned to a different cluster. Lastly, the distance

among the data instance and center of the cluster is calculated by combining the instance and cluster center probabilities and similarities. Then, the current instance is allocated to the nearby cluster centre using the computed distance.

The probability-based distance function is described in Algorithm 3.It computes the distance between every pair of attribute values.

Algorithm-3DistFn

Input: Dataset $D = \{ d_1, d_2, \ldots, d_n\}$

Output: Distance between two categorical value

```
1.  For i = 1 to A
2.    Get unique AttValue (AV)
3.    For x = 1 to AV
4.      For y = x+1 to AV
5.        Sum=0
6.        For j = 1 to A
7.          If (i != j)
8.            P1=Compute probability between x and y through j
9.            Sum=Sum+P1
10.         End If
11.       End For
12.       Sum=Sum/A
13.       Assign dist(x,y) =Sum
14.     End For
15.   End For
16. End For
```

Algorithm-4 explains the modified SOM clustering algorithm. Initially, the categorical attributes are converted into numerical attributes. Next, the similarity between two instances is calculated using the Euclidean distance.

Finally, this algorithm's learning rate and other parameters are updated based on the previous iteration.

Algorithm-4 MSOM

Input: Dataset *$X = \{ x_1, x_2, \ldots, x_n\}$, Number of Cluster k*

Output: *Clustering outcome*

1. Convert all categorical attributes into numerical attributes
2. Normalize all attribute values
3. Set k = 0, initialize the parameter for neighborhood Np, and assign the rate of learning $\mu = 1.0$;
4. Compute $W = \frac{\max(x)+\min(x)}{2} + \varepsilon \quad (\varepsilon = rand(0,1))$
5. For each instance in D
6. Find the distance between the instance and each weight vector
7. $dist(i,j) = \sqrt{\sum_{j=1}^{m}\left(CW_{ij} - x_i\right)^2}$
8. Update the weight $w_i(k+1) = \begin{cases} w_i(k) + \mu[x(k) - w_i(k)] & i \in N_p(k) \\ w_i(k) & i \notin N_p(k) \end{cases}$
9. Update parameters
10. Increment k
11. Terminated if the iteration number exceeds a specific threshold or the stopping criteria are met. Otherwise, return to step 4.

4.5 Experimental Results

Experiments are used in this part to assess how well the suggested task performs. The cluster results are examined using three openly accessible data sets as well as student data from seven questionnaires. Table 4.4 presents the overview of the experimental dataset.

Table 4. 4 Summary of Dataset

Dataset Type	Dataset	# Instances	# Numerical Features	# Categorical Features	# Classes
Student Info with Question Response	Emotional Intelligence (EIQ)	1000	2	11	3
	Eysenck Personality (EPQ)	1000	2	11	3
	General Self Efficacy (GSE)	1000	2	11	2
	Emotional Happiness (EH)	1000	2	11	3
	Positive /Negative Attitude (PNA)	1000	2	11	3
	Self Esteem (RSE)	1000	2	11	3
	Self Determination (SDS)	1000	2	11	2
Medical	Heart	293	7	6	5
	Dermatology	358	1	33	6

Purity, Rand Index (RI), Normalised Mutual Information (NMI), Precision (Pre), and Recall (Rec) are the measures used to evaluate clustering results. The classes-to-cluster assignment (CCA) , is presented in Table 4.5, which isused to calculate these evaluation measures.

Table 4. 5 Classes to Cluster Assignment Table

	C_1	C_2	….	C_k	Sum
P_1	a_{11}	a_{12}	….	a_{1k}	SP_1
P_2	a_{21}	a_{22}	….	a_{2k}	SP_2
⋮	⋮	⋮	⋮	⋮	⋮
P_c	a_{c1}	a_{c2}	….	a_{ck}	SP_c
Sum	SC_1	SC_2	….	SC_k	

Consider a dataset D = {D1, D2, D3,..., Dn} containing n instances, and let C = {C1, C2, ..., Ck} represent a k clusters set obtained from D utilizing a clustering algorithm. The set of c true classes of D, denoted as P = {P1, P2, ..., Pc}, is additionally considered. The number of common data points between Pi and Cj is represented by aij = |Pi ∩ Cj| in Table 2. SPi and SCj represent the number of data points in Pi and Cj, respectively. The evaluation metrics are calculated as follows.$Purity = \frac{1}{n}\sum_k \max_c |a_{kc}|$

$$ARI = \frac{\sum_{ij}\binom{a_{ij}}{2} - \left[\sum_i \binom{SP_i}{2}\sum_j \binom{SC_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{SP_i}{2} + \sum_j \binom{SC_j}{2}\right] - \left[\sum_i \binom{SP_i}{2}\sum_j \binom{SC_j}{2}\right]/\binom{n}{2}}$$

Here $\binom{n}{2} = n(n-1)/2$

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{k} a_{ij}\log\left(\frac{a_{ij}*n}{SP_i*SC_j}\right)}{\sqrt{\sum_{i=1}^{c} SP_i \log\left(\frac{SP_i}{n}\right)\sum_{j=1}^{k} SC_j log\left(\frac{SC_j}{n}\right)}}$$

$$Pre = \frac{1}{c}\sum_{i=1}^{c}\frac{\max_k a_{ki}}{SP_i}$$

$$Rec = \frac{1}{k}\sum_{i=1}^{k}\frac{\max_c a_{ci}}{SC_i}$$

The number of clusters detected during this experiment was equal to the number of classes in the dataset, i.e., c = k. Purity, RI, NMI, Pre, and Rec values with higher values suggest better clustering outcomes. Figure 4.1 displays the CCA for the EI dataset, demonstrating that the majority of the classes have been clustered accurately.

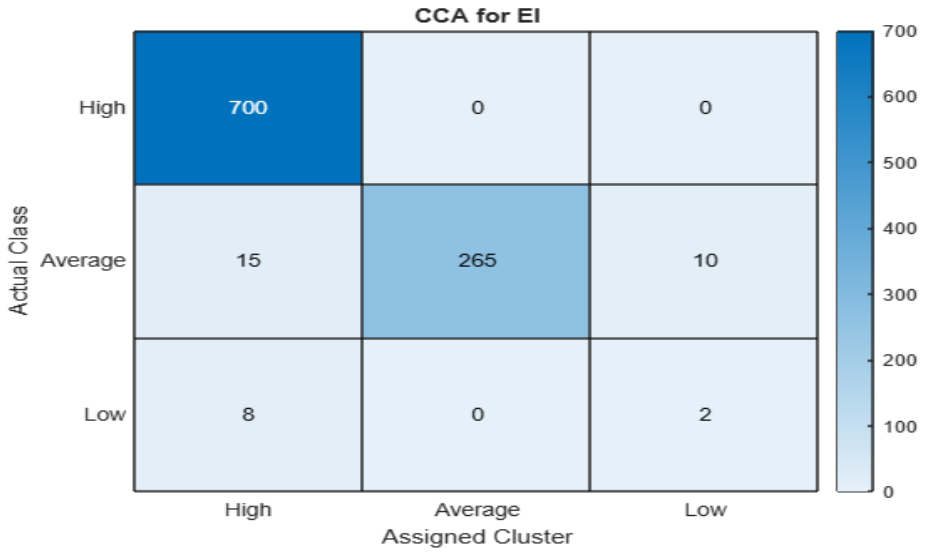


Figure 4. 1 Heatmap for EI

Figures 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7 show the EP, GSE, EHQ, PNA, RSE, and SDS heatmap chart.

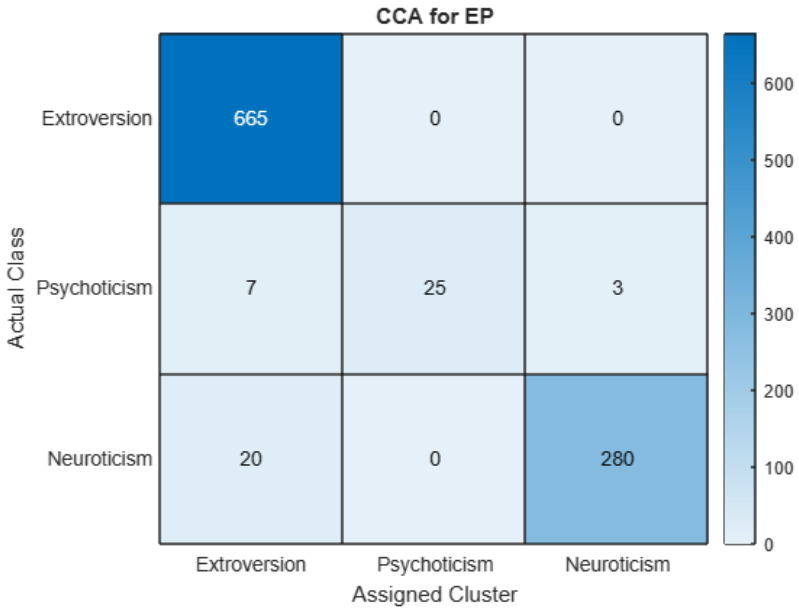


Figure 4. 2 Heatmap for EP

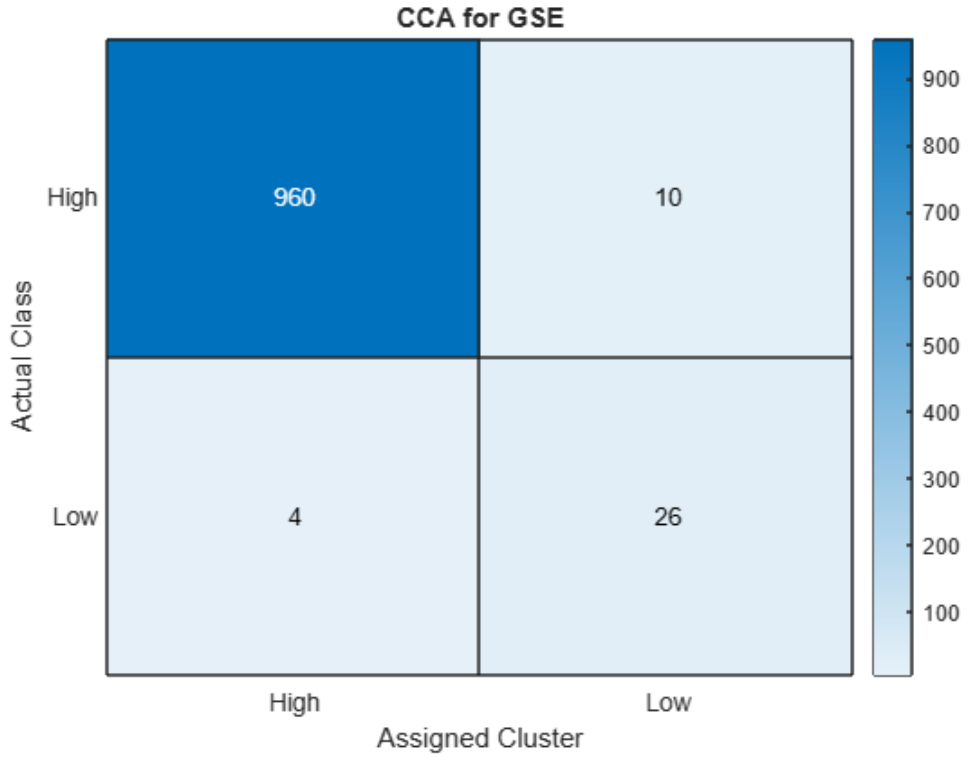


Figure 4. 3 Heatmap for GSE

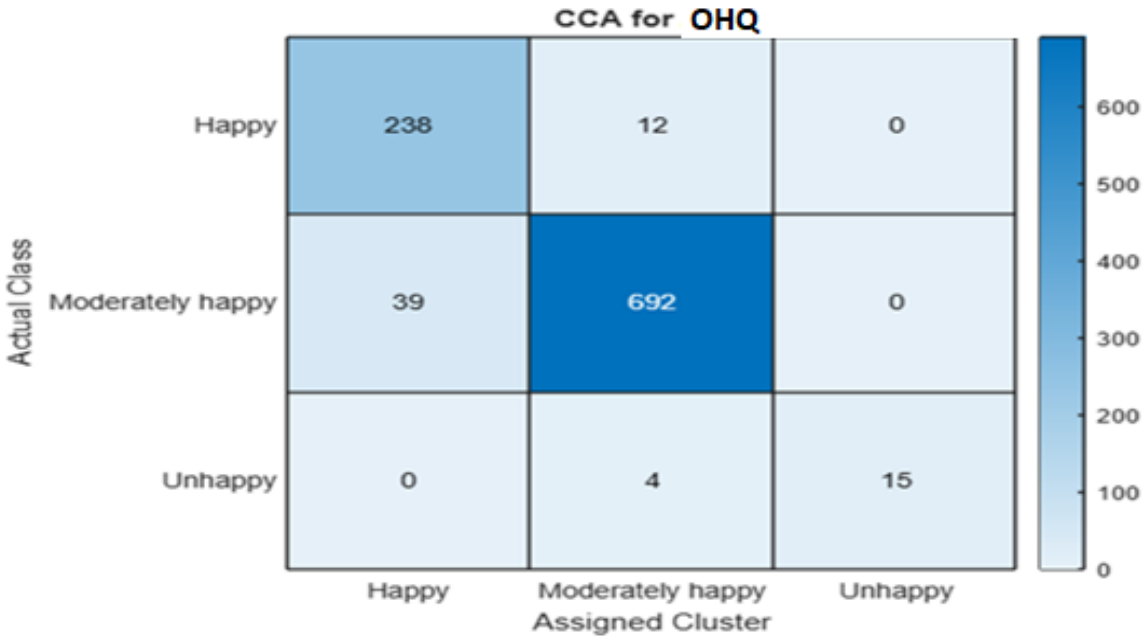


Figure 4. 4 Heatmap for OHQ

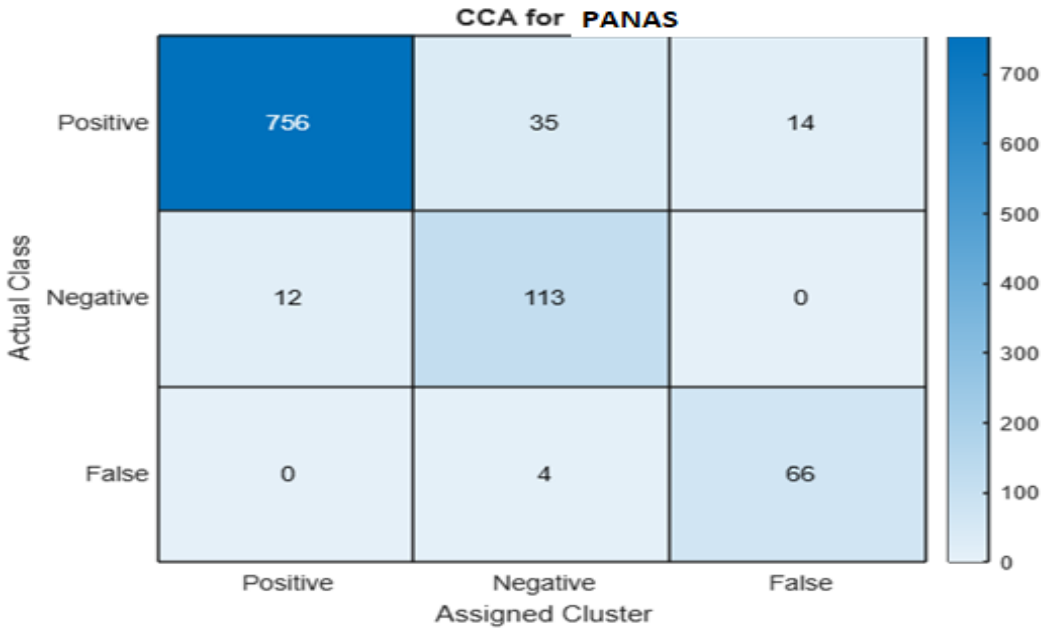


Figure 4. 5 Heatmap for PANAS

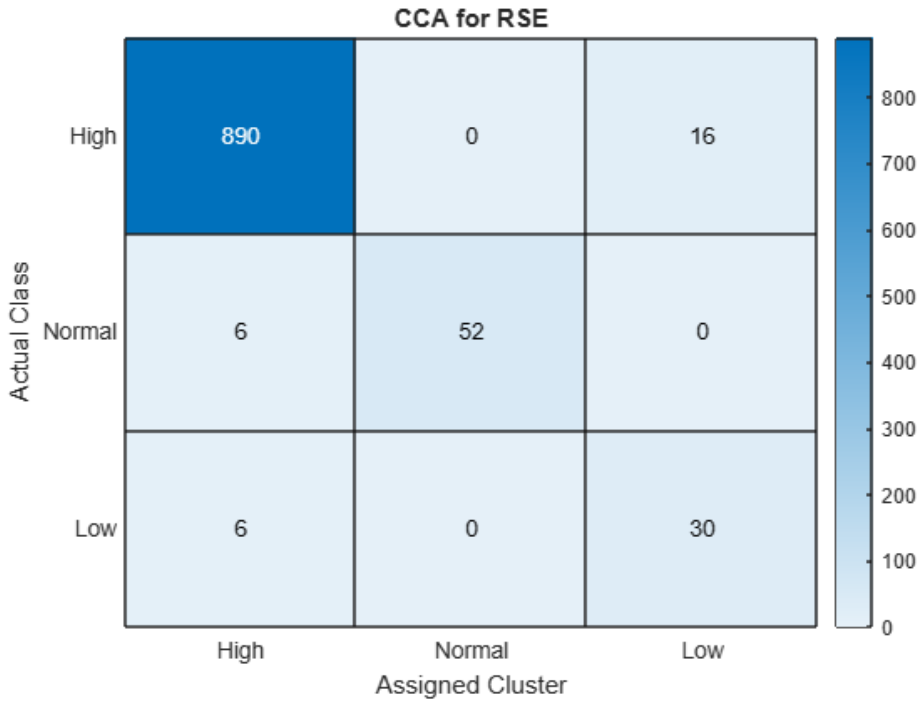


Figure 4. 6 Heatmap for RSE

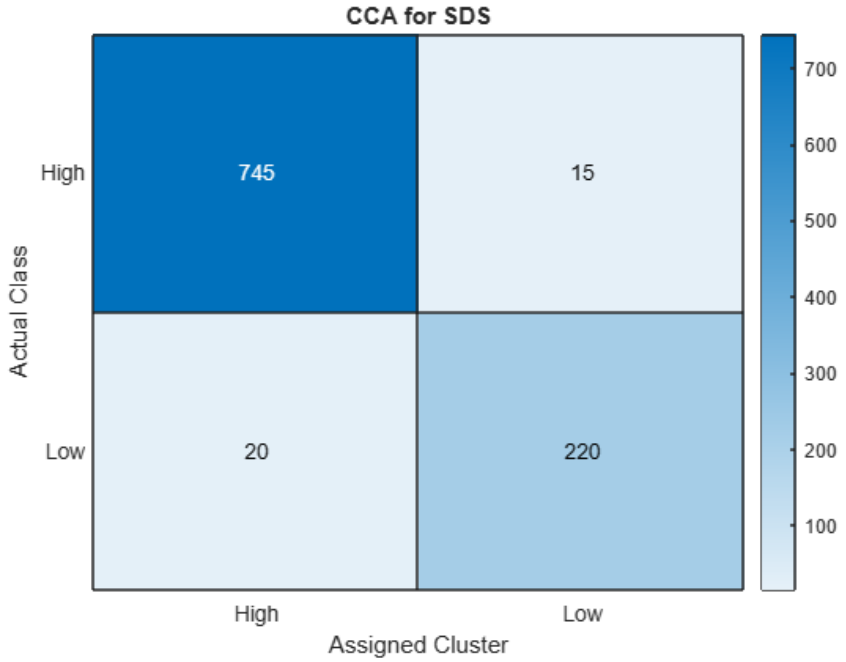


Figure 4. 7 Heatmap for SDS

Figure 4.8 compare evaluation metrics for different datasets.

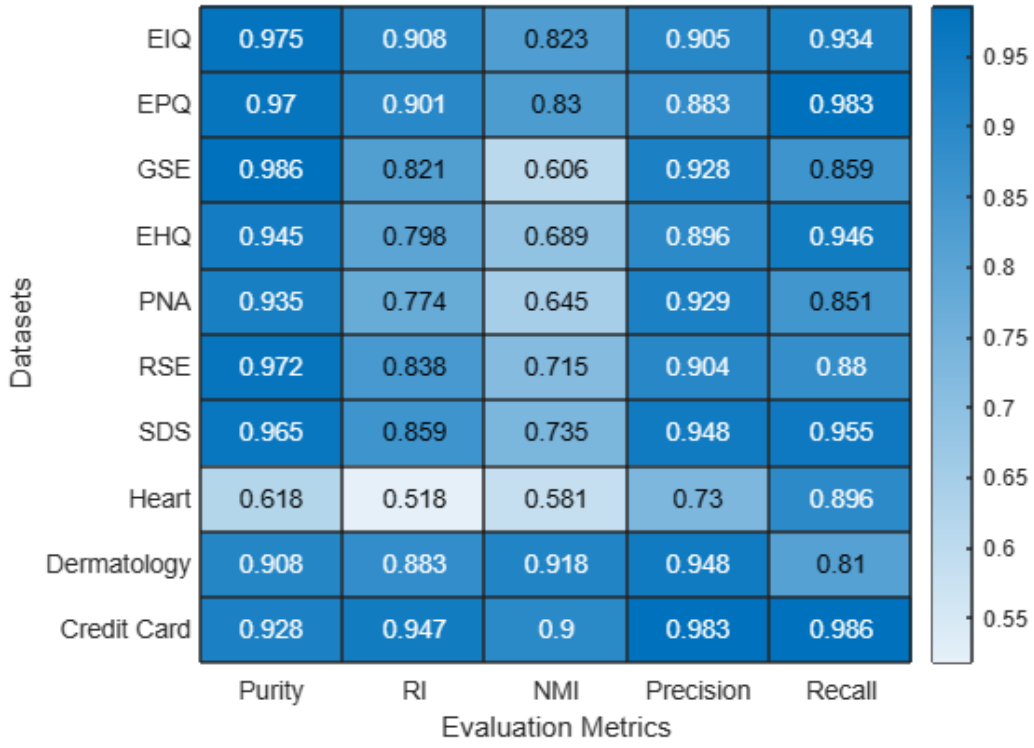
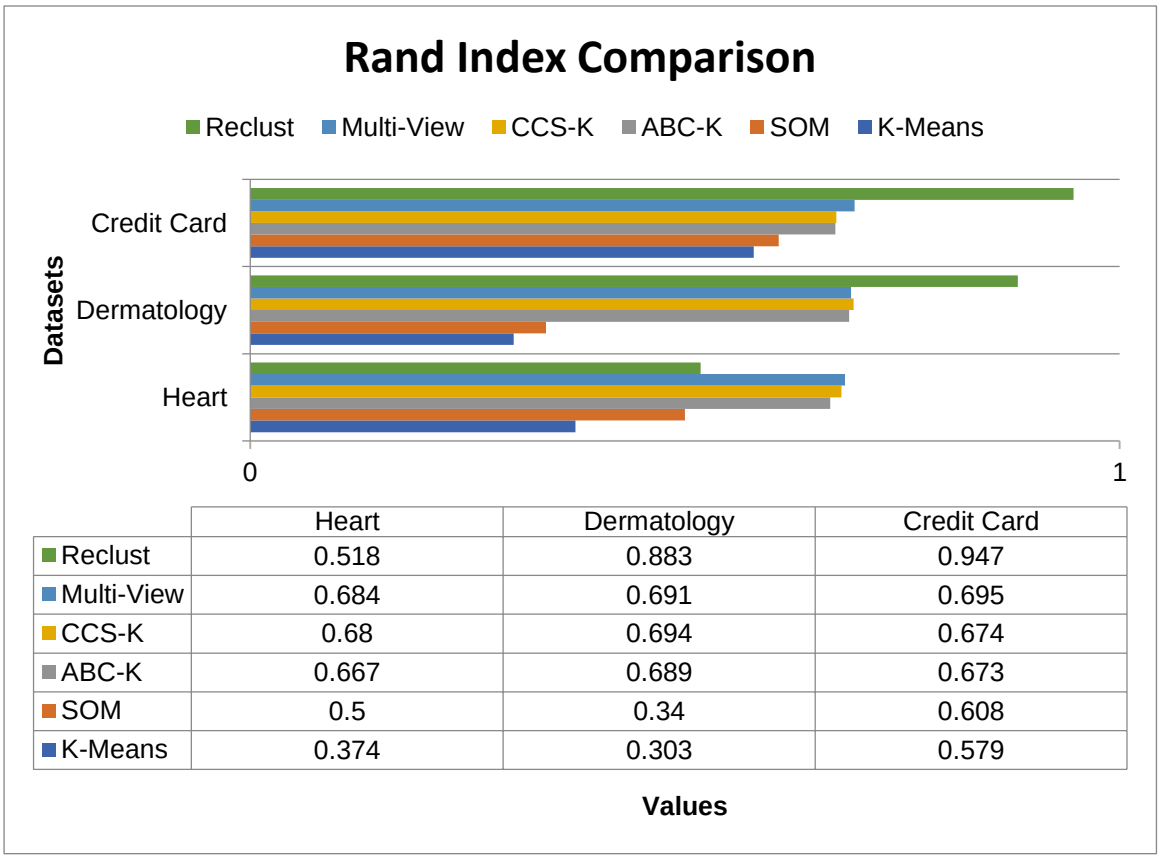


Figure 4. 8 Heatmap for evaluation metrics

The evaluation metrics RI, Precision, and Recall of the proposed Reclust algorithm are compared with those of K-means, SOM, ABC-K-Prototypes [171], CCS-K-Prototypes [148], and Multi-view K-Prototype [169]. Table 4.6 and Figure 4.9 present the comparison of Rand Index.

Table 4. 6 RI Comparison

Dataset	K-Means	SOM	ABC-K	CCS-K	Multi-View	Reclust
Heart	0.374	0.50	0.667	0.680	**0.684**	0.518
Dermatology	0.303	0.34	0.689	0.694	0.691	**0.883**
Credit Card	0.579	0.608	0.673	0.674	0.695	**0.947**



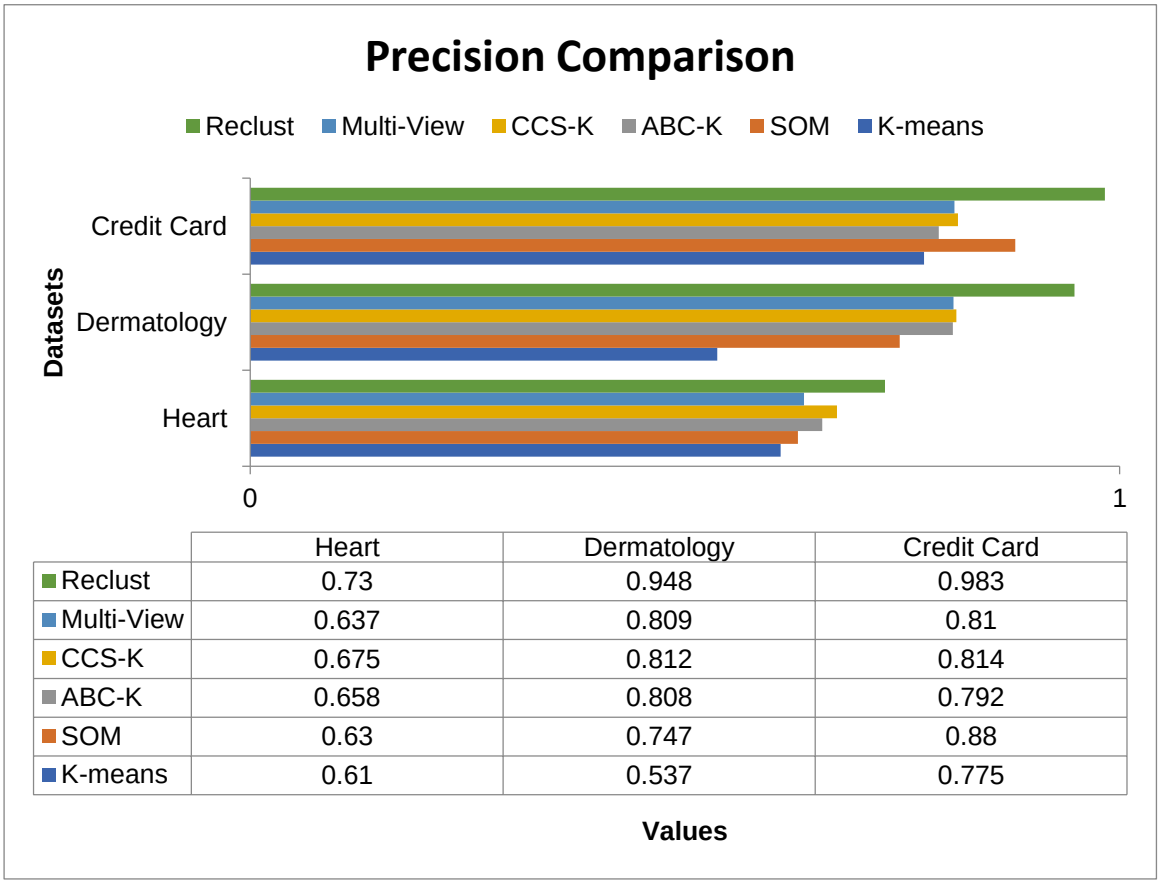
	Heart	Dermatology	Credit Card
Reclust	0.518	0.883	0.947
Multi-View	0.684	0.691	0.695
CCS-K	0.68	0.694	0.674
ABC-K	0.667	0.689	0.673
SOM	0.5	0.34	0.608
K-Means	0.374	0.303	0.579

Figure 4. 9 Comparison of Rand Index

Table 4.7 and Figure 4.10 display the precision comparison.

Table 4. 7 Precision Comparison

Dataset	K-means	SOM	ABC-K	CCS-K	Multi-View	Reclust
Heart	0.61	0.63	0.658	0.675	0.637	**0.730**
Dermatology	0.537	0.747	0.808	0.812	0.809	**0.948**
Credit Card	0.775	0.880	0.792	0.814	0.810	**0.983**



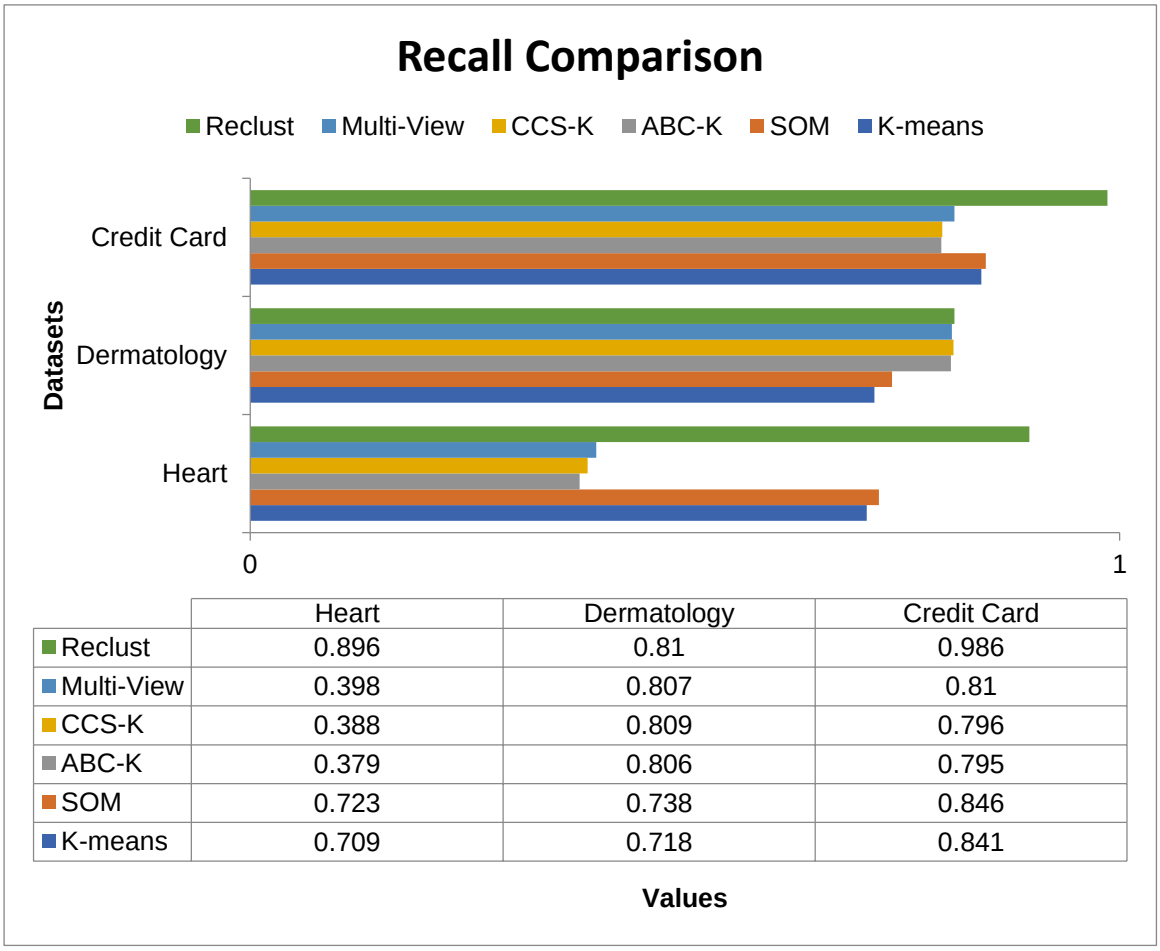
	Heart	Dermatology	Credit Card
Reclust	0.73	0.948	0.983
Multi-View	0.637	0.809	0.81
CCS-K	0.675	0.812	0.814
ABC-K	0.658	0.808	0.792
SOM	0.63	0.747	0.88
K-means	0.61	0.537	0.775

Figure 4. 10 Comparison of Precision

Table 4.8 and Figure 4.11 display the recall comparison.

Table 4.8 Recall Comparison

Dataset	K-means	SOM	ABC-K	CCS-K	Multi-View	Reclust
Heart	0.709	0.723	0.379	0.388	0.398	**0.896**
Dermatology	0.718	0.738	0.806	0.809	0.807	**0.810**
Credit Card	0.841	0.846	0.795	0.796	0.810	**0.986**



	Heart	Dermatology	Credit Card
Reclust	0.896	0.81	0.986
Multi-View	0.398	0.807	0.81
CCS-K	0.388	0.809	0.796
ABC-K	0.379	0.806	0.795
SOM	0.723	0.738	0.846
K-means	0.709	0.718	0.841

Figure 4. 11 Recall Comparison

4.6 Summary

Effective clustering of mixed numerical and categorical datasets is common due to the prevalence of mixed attributes in real-world data. An efficient clustering technique for organizing datasets containing a mix of numerical and categorical attributes is introduced in this study. Additionally, repetitive re-clustering and cluster validation techniques improve the quality of clustering outcomes. The reclust algorithm was evaluated using various datasets, and its performance was measured based on recall, NMI, precision, clustering purity, and rand index. The outcomes obtained from the experiments display the better performance of the reclust algorithm.