Predicting poor compliance to psychotropics using machine learning approach

by Dhrubang Utpal Talukdar
Under Guidance of
Dr. Rosy Sarmah
Dr. Siddeswara BL

Problem Definition

Objective:

- Cluster children based on psychotropic compliance patterns to identify groups with similar behavior.

Goals:

- Analyze factors influencing low compliance to understand and predict the level of compliance for each child.
- Develop a user friendly interface to be used by clinicians

PLANNING

DATA GENERATION

Since the data collection was still going on, we planned to generate synthetic data according to the requirements provided by the clinician.

CLUSTERING

Analyze and compare the various clustering algorithms based on various evaluation metrics.

CREATE APPLICATION

Use the best clustering algorithm and create an application to predict compliance on basis of various inputs.

DATA GENERATION

- This project is a collaborative project with Lokopriya Gopinath Bordoloi Regional Institute of Mental Health.

- **Aim** : To predict treatment compliance

- Initially synthetic dataset was generated with 79 columns in total (78 columns contain various details and 79th column contains the overall compliance of the child undergoing the treatment).

- Data collection is going on.

DATA GENERATION

Personal Information

It includes details like the child's name, age, gender, district of residence, family income, referral, the chief complaint, and information about their birth, such as weight, height, and head circumference.

Medical Information

We also have information about their family's medical history, past treatments, medication, the dose limit, the total cost, and the doctor's diagnosis. There are also measures of how well they followed their treatment plan, like mean gap ratio, medication possession ratio, total follow ups, the frequency of follow ups, maximum compliance period and total duration of medication.

Compliance

There's one more column that tells us how well the children followed their treatment, and it gives three results: Good, Bad, or Satisfactory.

IMPORTANT DEFINITIONS

Mean gap ratio

Total months of follow-ups divided by no. of follow-ups

Medication possession ratio

Total number of days when medications were taken divided by summation of total number of days when medications were taken with total off medication period

Follow up Frequency

Average follow up duration of patient in a year

DATA GENERATION

01

STAGE 1

A python code was written to generate the dataset with the help of lists and dictionaries.
We put random values in all the rows. A doctor was consulted for suggestions.

STAGE 2

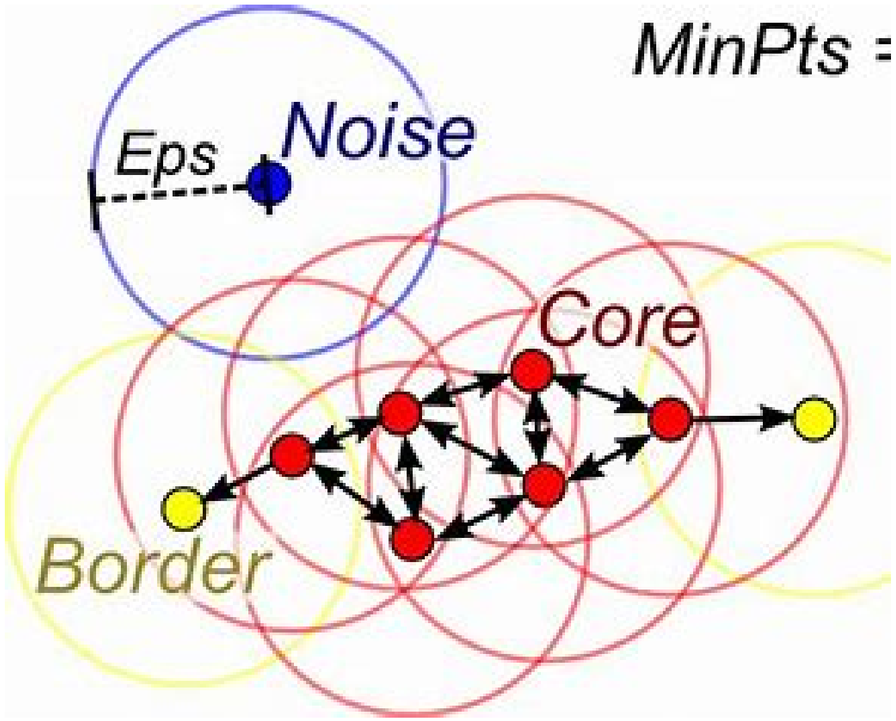
A few corrections were advised. Upon performing the corrections, a few checks followed through until we arrived at a dataset which was quite close to the original dataset verified by the doctor.

02

ALGORITHMS

- DBSCAN
- K-MEDOIDS
- FUZZY C-MEANS
- HIEARCHICAL-KNN
- ASSOCIATION-KNN

DBSCAN



01 DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm. It identifies clusters based on the density of data points in the feature space, distinguishing between dense regions and sparse areas.

02 It defines clusters through density reachability, connecting points within a specified radius and requiring a minimum number of neighbors to form dense regions utilizing parameters like epsilon (ε) and minPts

ADVANTAGES

- Robust to Noise and Outliers
- Can detect arbitrary shaped cluster

DISADVANTAGES

- Sensitive to input parameters
- Sensitive to cluster of varying densities

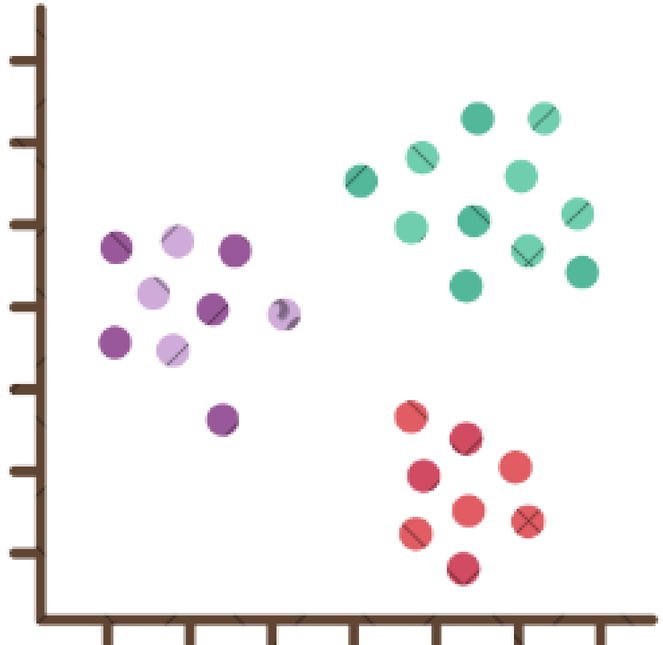
K-MEDOID

1

K-Medoid is a clustering algorithm that partitions a dataset into K clusters, where each cluster is represented by a medoid – the data point minimizing the average dissimilarity.

2

Initialization, Assignment, and Update Involves selecting initial medoids, assigning points to nearest medoids, and iteratively updating medoids for optimal clustering.



ADVANTAGES

- Easy to implement
- Faster and converges faster compared to k means

DISADVANTAGES

- It requires specifying the no of clusters(k) in advance
- It cannot handle noisy data & outliers.

FUZZY C-MEANS

Point 1

- Fuzzy C-Means is a clustering algorithm that assigns each data point to every cluster with a certain degree of membership, allowing for soft, overlapping cluster assignments.

Point 2

- Unlike traditional clustering algorithms, FCM assigns fuzzy memberships to each data point, indicating the likelihood of belonging to multiple clusters simultaneously.

Point 3

- The algorithm minimizes the objective function, which considers the distances between data points and cluster centers weighted by fuzzy memberships.

Advantages

- FCM provides soft cluster assignments, allowing data points to belong to multiple clusters simultaneously, making it flexible
- Fuzzy memberships make FCM less sensitive to outliers and noise, enhancing its ability to handle datasets with irregularities.

Disadvantages

- Sensitive to input parameters
- FCM involves iterative optimization, which can be computationally intensive

ASSOCIATION-KNN

- The Association KNN algorithm considers the k nearest points for each point within the same cluster.

- It assigns points to clusters starting from the cluster with the least average intra cluster distance.

- It continues adding all the points to this cluster if they lie within the threshold distance and have not been assigned earlier.

- If no new points can be assigned to the cluster, the process moves on to the next smallest cluster, and this continues until all points are assigned to a cluster. This is also a hard clustering algorithm where one point is assigned to one cluster only.

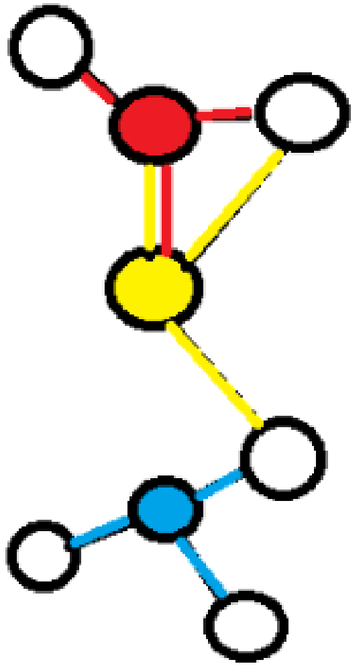
ASSOCIATION-KNN

Algorithm

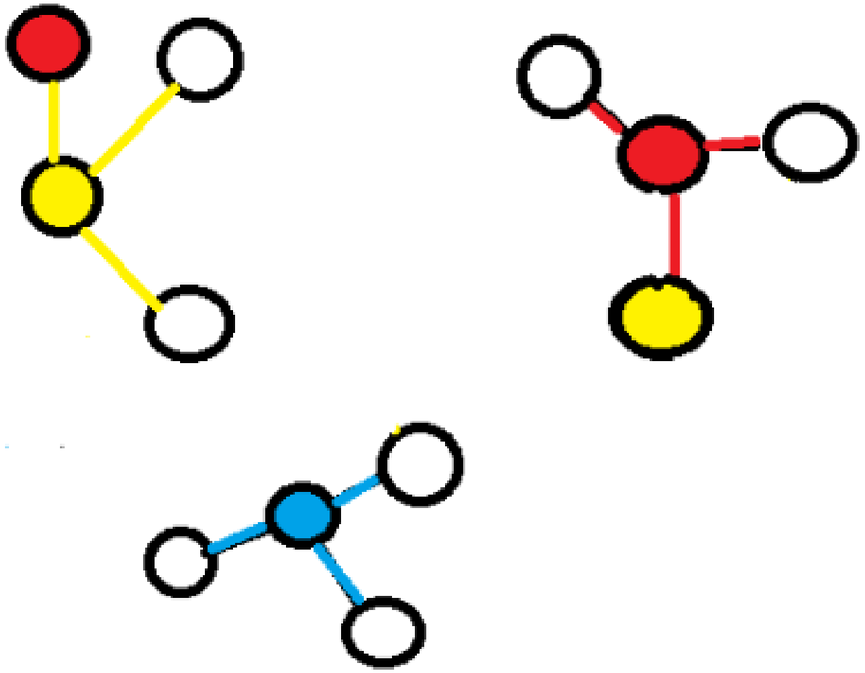
1) Input the value of k and threshold
2) Compute the distance matrix
3) Store the computed distance matrix in another variable
4) Sort each row of the distance matrix in ascending order with indexes
5) Calculate the average cluster distance for the points with its k nearest neighbour, which are the first k values of the sorted distance matrix and store it
6) Sort the average cluster distances with its position
7) Assign default clusterid of 1 to all the points which form the smallest cluster
8) Initialize clusternumber to 1
9) Repeat step 10 to step 15 till clusterid != clusterid1
10) Copy clusterid values to clusterid1
11) Make a cluster for the current clusternumber under observation using clusterid
12) Calculate the average cluster distance of the point with the current cluster
13) If the point under observation is within the threshold value and not assigned to any other cluster then append point to the current cluster under observation using clusterid1
14) Exchange values of clusterid and clusterid1
15) Go to step 9
16) Go to step 20 if all points have been checked once
17) Now move to next smallest cluster and continue till we find an unassigned point
18) Make the next cluster to start observation with which is the current chosen point and all its k nearest point which are unassigned
19) Go to step 9
20)If a cluster contains less number of points than the outlier threshold then label the cluster as an outlier
21) Exit

input k,threshold

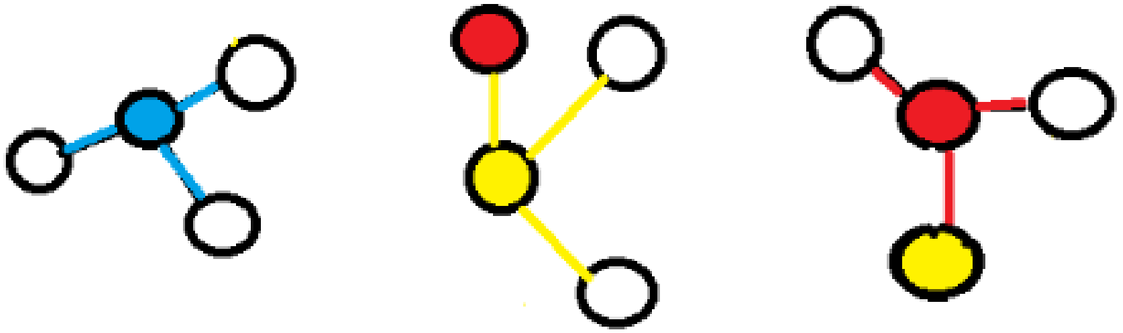
Compute/input distance matrix



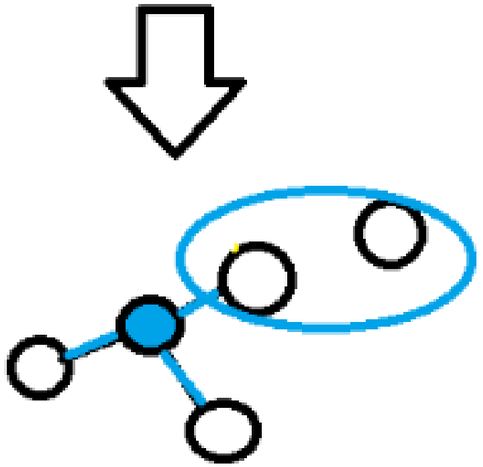
Compute k nearest for each pont



Compute average intracluster distance of all points with k nearest neighbours



Sort the cluster according to the calculated distances in increasing order

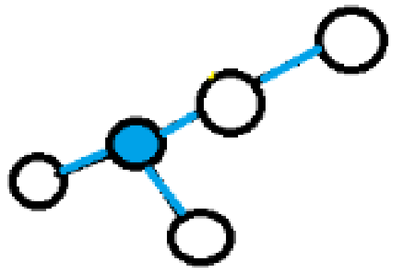


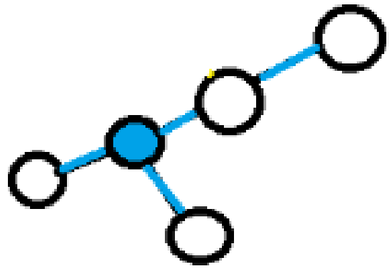
<= threshold

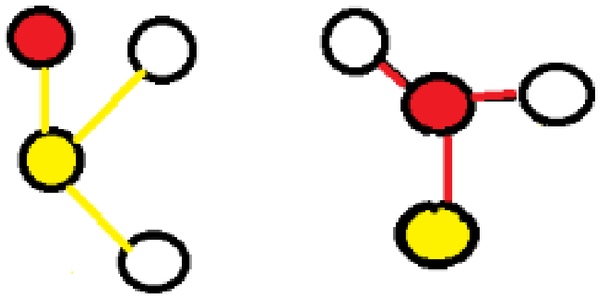
If a point is within the threshold include the point in the cluster

Include all points within threshold to that cluster and its unassigned

Repeat until the cluster converges and no new point can be added







Now begin again with the cluster having the next smallest intracluster distance

If the point is already assigned to any previous cluster we move to the next point

Repeat until all points processed

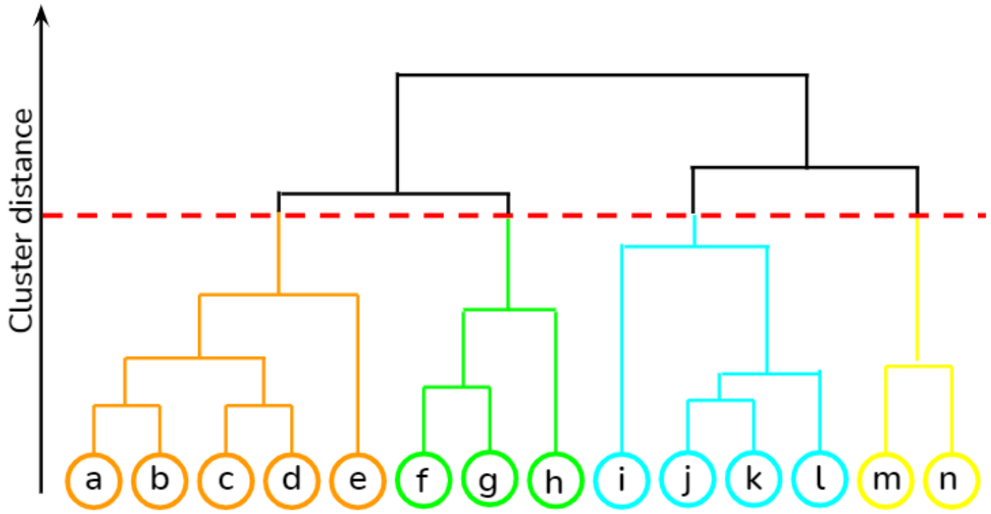
ADVANTAGES

1. No prior information about the number of clusters required.
2. It is a faster algorithm compared to hierarchical approach

DISADVANTAGES

1. Sensitive to user input
2. Algorithm cannot undo what was previously done

HIERARCHICAL-KNN



1. Hierarchical KNN, considers the k nearest points for each point within the same cluster

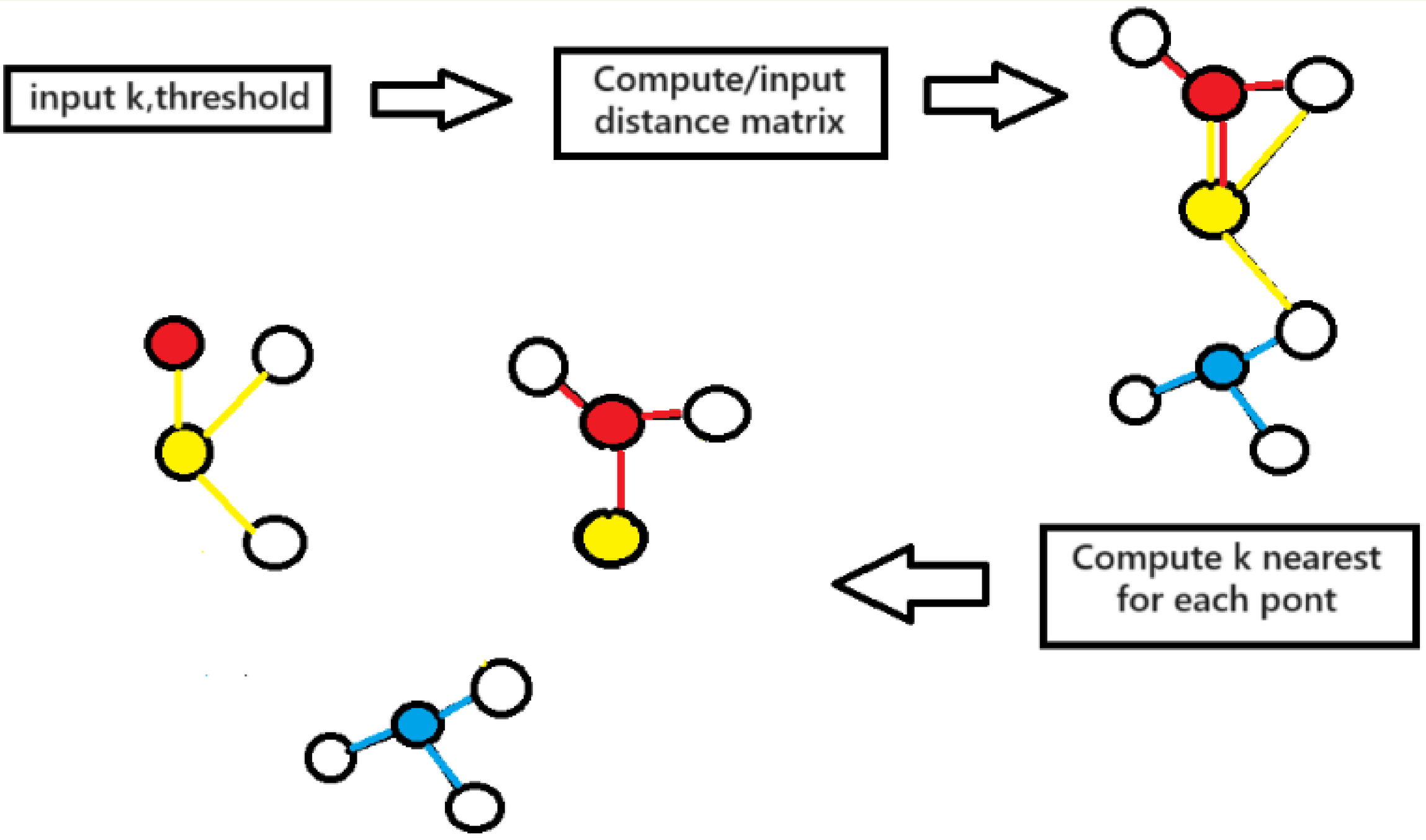
2. It merges clusters in an agglomerative manner until the condition that the average cluster distance between two clusters remains less than or equal to the threshold

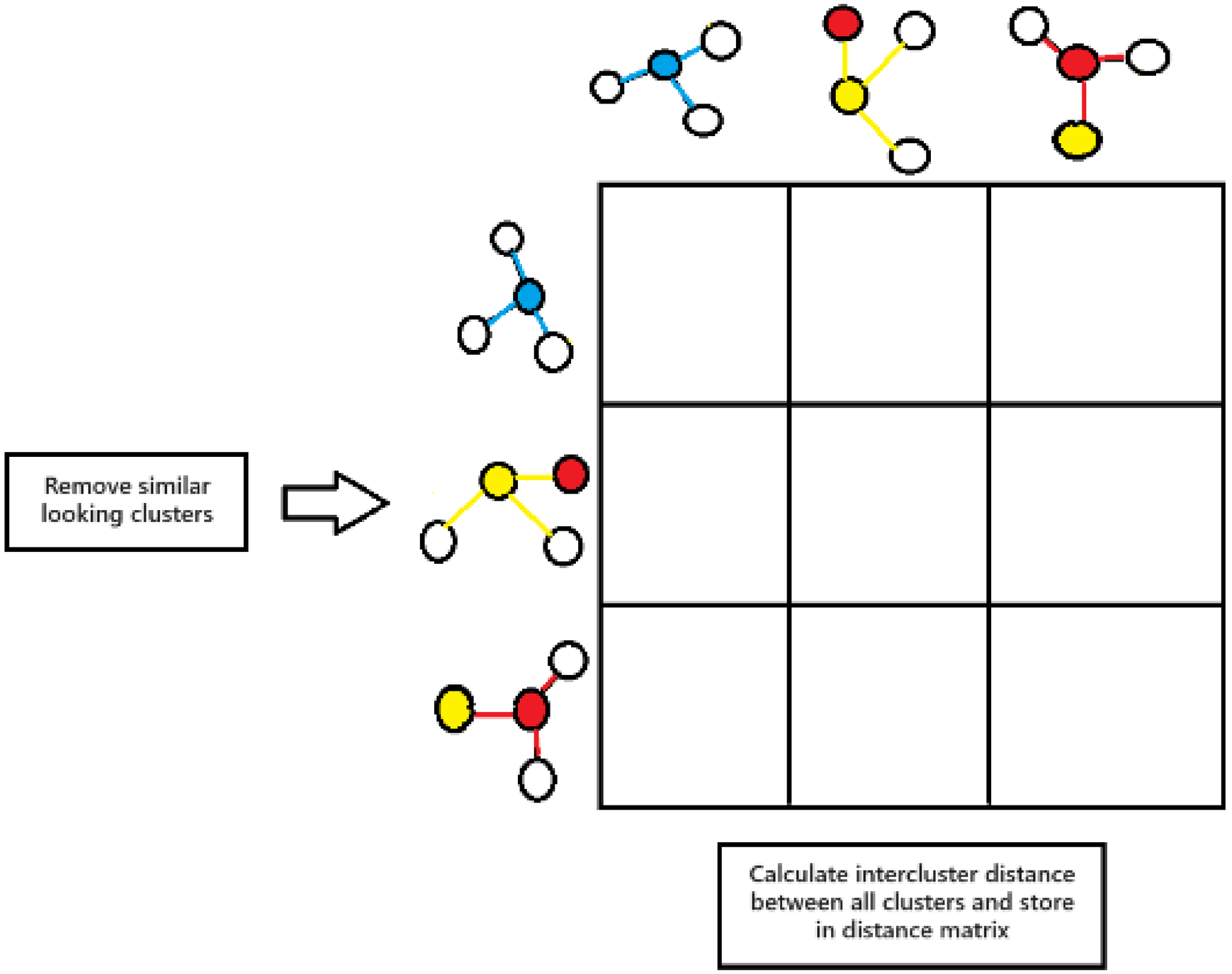
3. The algorithm is a hard clustering algorithm and assigns only one point to one algorithm

HIERARCHICAL-KNN

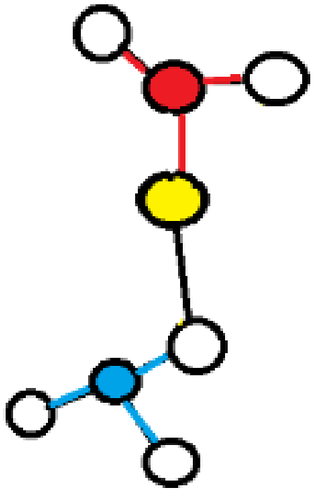
Algorithm

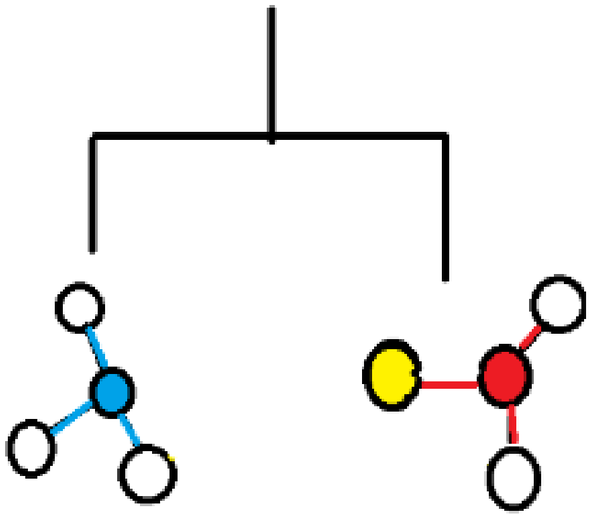
1) Input the value of k, threshold and outlier threshold
2) Compute the distance matrix
3) Store the computed distance matrix in another variable
4) Sort each row of the distance matrix in ascending order with indexes
5) Calculate the average cluster distance for the points with its k nearest neighbour, which are the first k values of the sorted distance matrix and store it
6) Sort the average cluster distances with its position
7) Create a dummy list which consists of all the k nearest points along with the points it is near to
8) Remove similar looking clusters
9) Calculate the average cluster distance between 2 clusters and store it in a 2d list
10) Get the row number and column number from the distance matrix where value is below threshold
11) If no such value found which is below threshold then go to step 15
12) Merge both clusters and remove similar points
13) Remove the clusters which were merged
14) Go to step 8
15) Sort the clusters we get to start from largest cluster
16) Assign clusterid to points according to cluster
17)If a cluster contains less number of points than the outlier threshold then label the cluster as an outlier
18) Exit


input k,threshold
Compute/input distance matrix
Compute k nearest for each pont


Remove similar looking clusters
Calculate intercluster distance between all clusters and store in distance matrix

Append this cluster to the cluster list and remove the clusters which were merged to form it



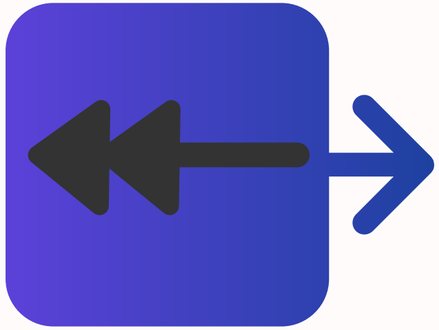


Repeat the steps till convergence

Pick the smallest value in the distance matrix for different cluster and merge the two clusters if distance is within threshold and core points are unassigned

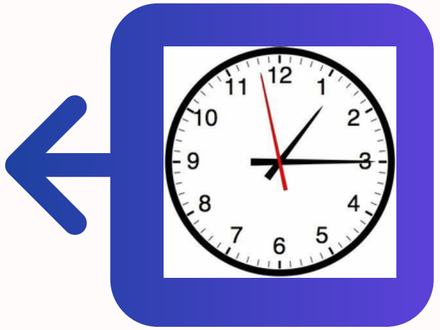
Advantages

- No prior information about the number of clusters required

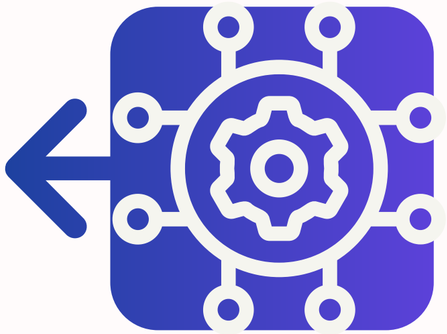


- Easy to implement

Disdvantages



- Time complexity is too much



- Algorithm cannot undo what was previously done

CONVERT TO MACHINE READABLE FORM

01

NOMINAL VARIABLES

"How is dissimilarity computed between objects described by nominal attributes?" The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p},$$

where m is the number of *matches* (i.e., the number of attributes for which i and j are in the same state), and p is the total number of attributes describing the objects.

Nomalize the feature values between 0 and 1

02

ORDINAL VARIABLES

Transform ordinal attribute values to ranks, r_{if}, with M_f ordered states.

Normalize ranks to [0.0, 1.0] for equal weighting using $z_{if} = \frac{r_{if} - 1}{M_f - 1}$.

03

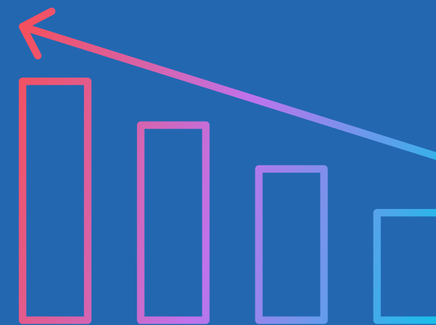
NUMERIC/RATIO VARIABLES

No changes needed

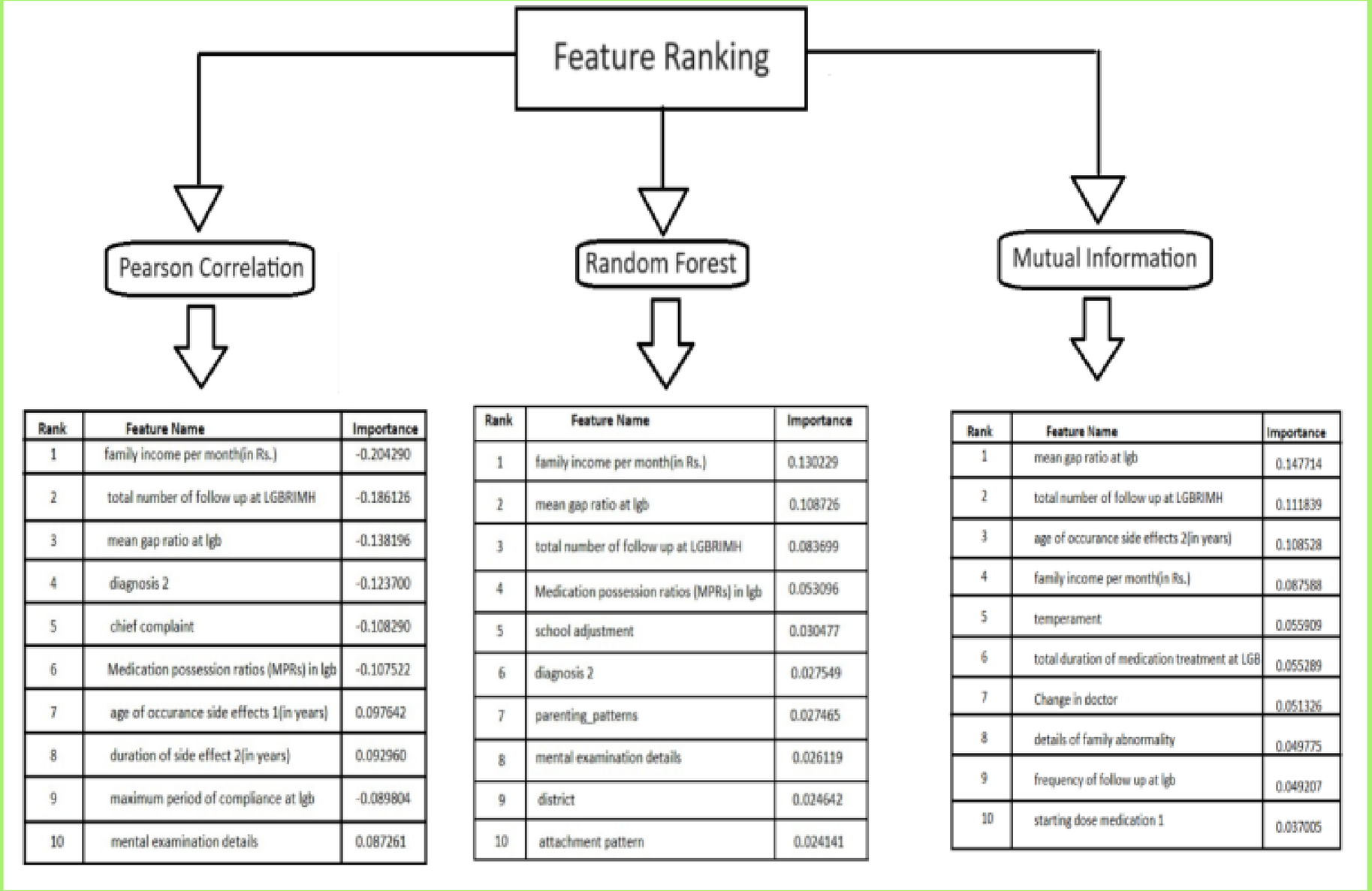
FEATURE RANKING

We have ranked the features which show the most similarities with the compliance column. We have done this using three feature selection measures – Correlation, Mutual Information and Random Forest Regressor

We find the rank of each feature for each of the similarity measure. Then we rank the features by computing the average rank which is the average of the rank of a feature in each similarity measure.



FEATURE RANKING



Rank	Feature Name	Importance
1	family income per month(in Rs.)	-0.204290
2	total number of follow up at LGBRIMH	-0.186126
3	mean gap ratio at lgb	-0.138196
4	diagnosis 2	-0.123700
5	chief complaint	-0.108290
6	Medication possession ratios (MPRs) in lgb	-0.107522
7	age of occurance side effects 1(in years)	0.097642
8	duration of side effect 2(in years)	0.092960
9	maximum period of compliance at lgb	-0.089804
10	mental examination details	0.087261

Rank	Feature Name	Importance
1	family income per month(in Rs.)	0.130229
2	mean gap ratio at lgb	0.108726
3	total number of follow up at LGBRIMH	0.083699
4	Medication possession ratios (MPRs) in lgb	0.053096
5	school adjustment	0.030477
6	diagnosis 2	0.027549
7	parenting_patterns	0.027465
8	mental examination details	0.026119
9	district	0.024642
10	attachment pattern	0.024141

Rank	Feature Name	Importance
1	mean gap ratio at lgb	0.147714
2	total number of follow up at LGBRIMH	0.111839
3	age of occurance side effects 2(in years)	0.108528
4	family income per month(in Rs.)	0.087588
5	temperament	0.055909
6	total duration of medication treatment at LGB	0.055289
7	Change in doctor	0.051326
8	details of family abnormality	0.049775
9	frequency of follow up at lgb	0.049207
10	starting dose medication 1	0.037005

AVERAGE RANK

Rank	Feature	Average_score
1	family income per month(in Rs.)	2
2	mean gap ratio at lgb	2
3	total number of follow up at LGBRIMH	2.333333333
4	Medication possession ratios (MPRs) in lgb	10
5	diagnosis 2	12.66666667
6	parenting_patterns	13.33333333
7	details of family abnormality	15
8	school adjustment	15
9	mental examination details	16.66666667
10	past medical history	17.33333333
11	total duration of medication treatment at LGB(in years)	18.33333333
12	temperament	18.66666667
13	frequency of follow up at lgb	22.33333333
14	chief complaint	22.66666667
15	maximum period of compliance at lgb	22.66666667
16	age of occurance side effects 2(in years)	23.33333333
17	cost of medication	23.33333333

ASSIGNING WEIGHTS

- 17 TOP FEATURES WERE SELECTED OUT OF THE TOTAL OF 78 FEATURES ACCORDING TO THEIR RANKING FOR CALCULATION OF THE DISSIMILARITY FEATURES.

- DIFFERENT WEIGHTS WERE ASSIGNED USING TRIAL AND ERROR METHOD TO THE FEATURES TO GET THE DISSIMILARITY MATRIX GIVING BEST RESULTS.

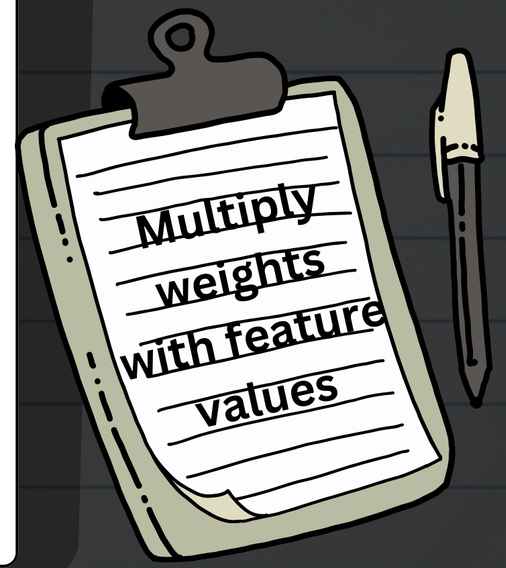
ASSIGN WEIGHTS

Rank	Feature	Weight
1	family income per month(in Rs.)	30
2	mean gap ratio at lgb	12
3	total number of follow up at LGBRIMH	15
4	Medication possession ratios (MPRs) in lgb	15
5	diagnosis 2	1
6	parenting_patterns	2
7	details of family abnormality	1
8	school adjustment	1
9	mental examination details	2
10	past medical history	1
11	total duration of medication treatment at LGB(in years)	1
12	temperament	1
13	frequency of follow up at lgb	1
14	chief complaint	0
15	maximum period of compliance at lgb	1
16	age of occurance side effects 2(in years)	1
17	cost of medication	1

HIERARCHICAL KNN

Rank	Feature	Weight
1	family income per month(in Rs.)	39
2	mean gap ratio at lgb	28
3	total number of follow up at LGBRIMH	23
4	Medication possession ratios (MPRs) in lgb	20
5	diagnosis 2	14
6	parenting_patterns	3
7	details of family abnormality	1
8	school adjustment	4
9	mental examination details	1
10	past medical history	3
11	total duration of medication treatment at LGB(in years)	3
12	temperament	1
13	frequency of follow up at lgb	1
14	chief complaint	0
15	maximum period of compliance at lgb	1
16	age of occurance side effects 2(in years)	1
17	cost of medication	1

ASSOCIATION KNN



DISSIMILARITY MATRIX

- We use Manhattan distance to find dissimilarities between each values of the features
- We get the dissimilarity matrix for each feature. Perform weighted addition to get the final dissimilarity matrix.
- Use this dissimilarity matrix as input for the particular clustering algorithm

DISSIMILARITY MATRIX

$$d = \sum_{i=1}^{p} d_i^{(f)} \delta_i^{(f)}$$

$$d_i^{(f)} = 0 \quad (\text{if } d_i^{(f)} = \varnothing)$$

, else the value calculated

$$\delta_i^{(f)} = 1 \quad (\text{if } \delta_i^{(f)} = \varnothing)$$

, else the value calculated

Evaluation Metrics

ARI

Adjusted Rand Index (ARI):

- **Definition:** ARI measures the similarity between true and predicted clusters while accounting for chance. It considers the ratio of agreements and disagreements between clusters, providing a normalized score that ranges from -1 to 1.
- **Formula:** $ARI = \frac{\text{RI} - \text{Expected Random Agreement}}{\text{Max Possible Agreement} - \text{Expected Random Agreement}}$

AMI

Adjusted Mutual Information (AMI):

- **Definition:** AMI quantifies the information shared between true and predicted clusters, considering both entropy and mutual information. A higher AMI score indicates better alignment between clusters.
- **Formula:** $AMI = \frac{\text{MI} - \text{Expected Mutual Information}}{\text{Average Entropy}}$

EVALUATION METRICS

ACCURACY

Accuracy is a widely used metric for evaluating the performance of classification models. It measures the proportion of correctly classified instances out of the total instances. The formula for accuracy is:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

SILHOUETTE INDEX

The silhouette index assesses the quality of clustering in unsupervised machine learning. It measures how well-separated clusters are and ranges from -1 to 1. A high silhouette index indicates well-defined clusters. The formula for silhouette index for an individual data point i is:

$$\text{Silhouette}_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Where:

- a_i is the average distance from the i-th point to other points in the same cluster.
- b_i is the average distance from the i-th point to points in the nearest cluster

EVALUATION METRICS

CALINSKI-HARABASZ

Definition: The Calinski-Harabasz index assesses clustering quality by quantifying the ratio of between-cluster variance to within-cluster variance.

Formula:

$$\text{Calinski-Harabasz Index} = \frac{\text{Between-Cluster Variance}}{\text{Within-Cluster Variance}} \times \frac{N-k}{k-1}$$

Interpretation: A higher Calinski-Harabasz index suggests well-defined, distinct clusters among the data points.

DAVIES-BOULDIN

$$DB = \frac{1}{k}\Sigma_{i=1}^{k} max(\frac{\Delta(X_i)+\Delta(X_j)}{\delta(X_i,X_j)})$$

where:

ΔX_k is the intracluster distance within the cluster X_k.

$\delta(X_i, X_j)$ is the intercluster distance between the clusters $X_i \; and \; X_j$.

RESULTS

ALGO	K	THRES HOLD	EPS	MIN POINT	OUTLIER THRESHOLD	ACC	ARI	AMI	SH	DB	CH
H-KNN	50	5	-	–	10	68.3%	0.24	0.26	0.48	3.7	60.2
A-KNN	50	0.5	-	–	5	65.4%	0.23	0.25	0.08	2.3	47.23
K-MEDOID	3	-	-	-	-	58%	0.23	0.30	0.11	2.18	60.01
FUZZY C-MEAN	3	-	-	-	-	45%	0.1	0.1	0.05	3.23	23.78
DBSCAN	-	-	30	4	4	49.7%	0.2	0.11	0.1	3.05	24.79

MODEL BUILDING

General Overview

Clusters given by hierarchical-KNN were used to build the webtool. The parameters used for Hierarchical-KNN were k - 50, threshold - 5 and outlier threshold - 10.

Weight

The entered parameters are converted to real numbers between 0 to 1 using method of mapping for nominal /ordinal attributes. Numeric/ratios are normalized between 0 and 1.This is multiplied by the weights given during training.

Prediction

Euclidean distance was used to assign the point to the nearest cluster. The prediction returned is the label of the cluster.

Tools Used

HTML,JavaScript, Flask(Python),CSS

Confusion Matrix for Synthetic Data

- Weighted Sensitivity:0.6825
- Weighted Precision:0.6832
- Weighted Accuracy: 0.7669
- Weighted F1 Score: 0.6825

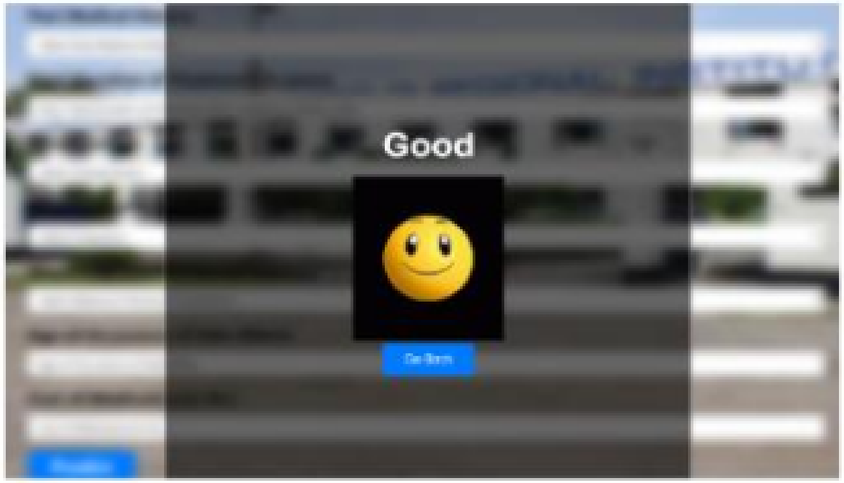
		Actual		
		Good	Satisfactory	Poor
Predicted	Good	49	35	0
	Satisfactory	28	118	26
	Poor	0	25	79

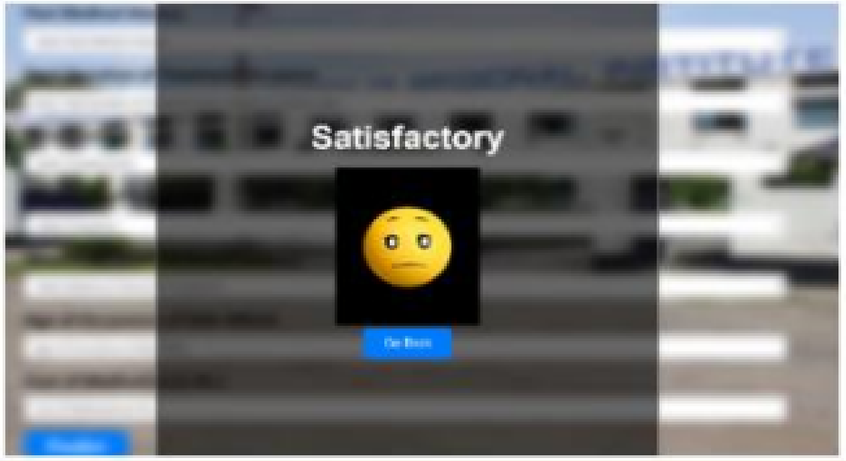
Confusion Matrix for Actual Data

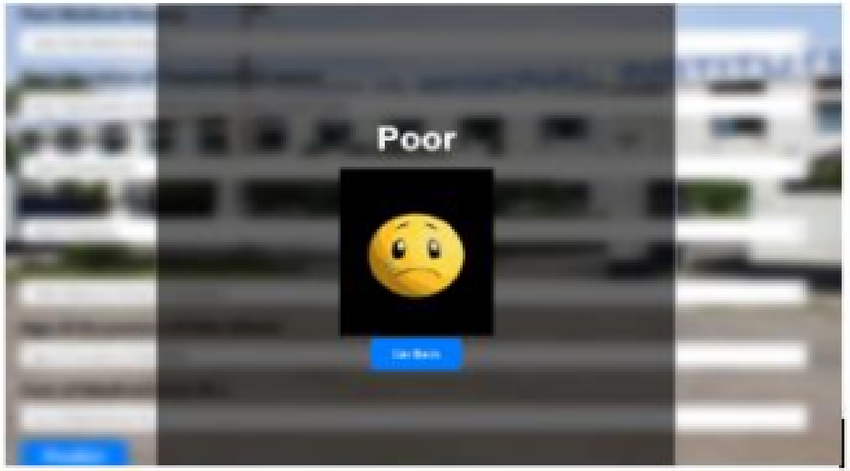
- Weighted Sensitivity:0.8010
- Weighted Precision:0.6071
- Weighted Accuracy: 0.6990
- Weighted F1 Score: 0.6048

		Actual		
		Good	Satisfactory	Poor
Predicted	Good	8	0	0
	Satisfactory	5	7	5
	Poor	1	0	2

DEMONSTRATION

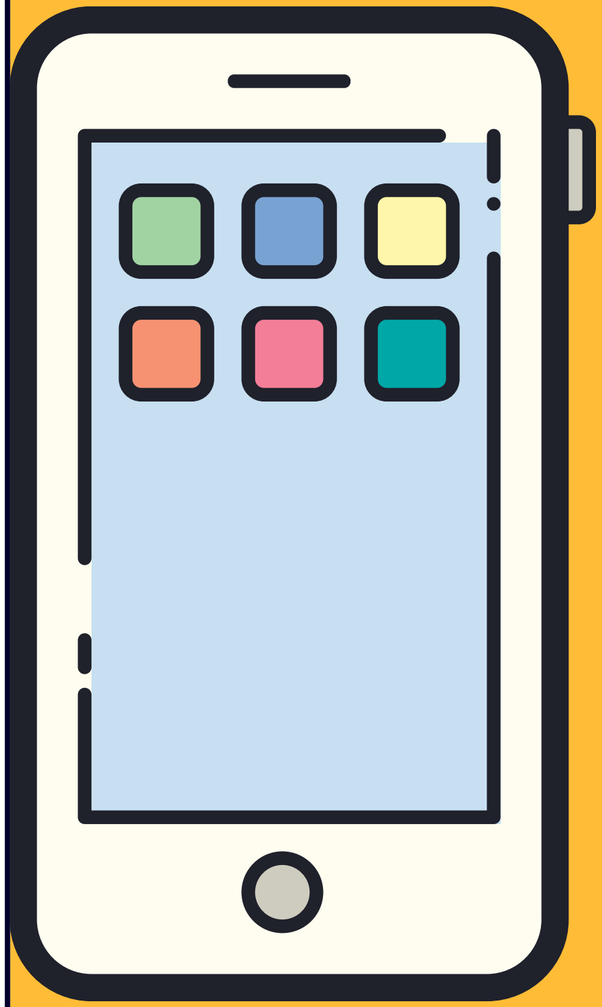






FUTURE WORKS

- Inclusion of more real time data
- Better measures to check belonginess of point to a cluster
- Supervised machine learning methods will be used
- We would try to improve the accuracy and improvise on the website
- An app will be developed



THANK YOU