

Unsupervised learning - Clustering

Learning Objectives

- Introduction to Unsupervised learning
- Types of clustering and distance calculation
- Euclidean and Non Euclidean distance
- K-means clustering
- Elbow method
- Visual analysis and Dynamic Clustering
- Hands on exercise on K-means clustering
- Case study

Introduction to Unsupervised learning



- Unsupervised Learning is a class of Machine Learning techniques to find the patterns in data.
- The data given to unsupervised algorithm are not labelled, which means only the input variables(X) are given with no corresponding output variables.
- Unsupervised learning is the training an algorithm using information that is neither classified nor labelled.

What is unsupervised learning? **greatlearning**

- No defined dependent and independent variables.
- Patterns in the data are used to identify / group similar observations

Clustering

- Clustering is primarily an exploratory technique to discover hidden structures of the data, possibly as a prelude to more focused analysis or decision process
 - A way to decompose a data set into subsets with each subset representing a group with similar characteristics.
 - Group such that objects in the same group are more similar to each other in some sense than to objects of different groups.
 - The groups are known as clusters and each cluster gets distinct label called cluster ID, the centroid of cluster.

What is clustering?

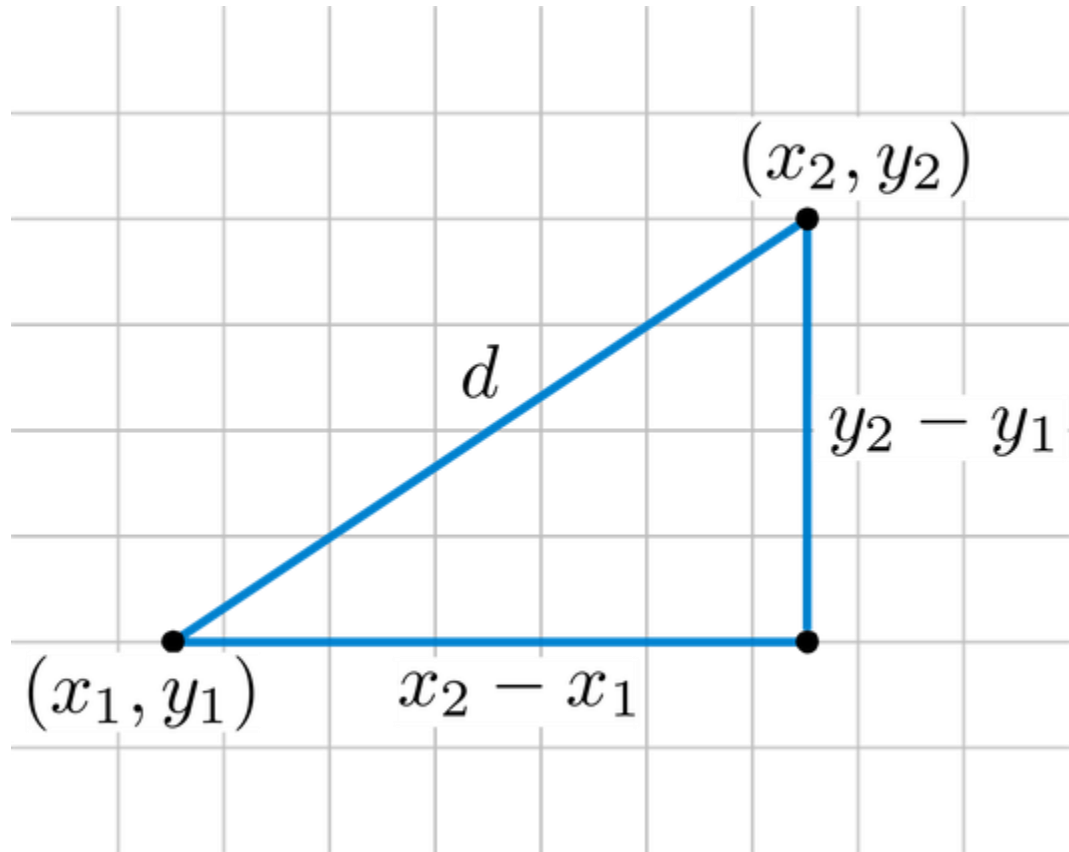
- task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups
- Objective - to ensure that the distance between datapoints in a cluster is very low compared to the distance between 2 clusters.

Clustering distances

1. Manhattan distance
2. Euclidean distance
3. Chebyshev distance

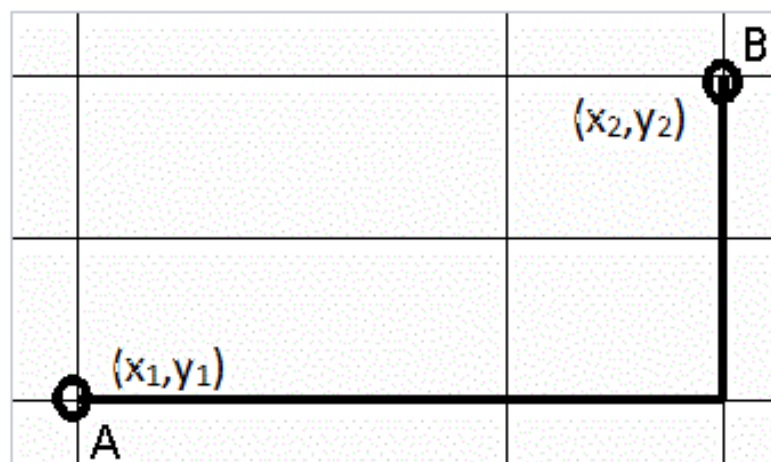
General use case for all the three distance measures above:
Minkowski distance

What is Euclidian distance?



What is Manhattan distance?

For a 2 – *dimensional* space. Distance between A and B is given by :



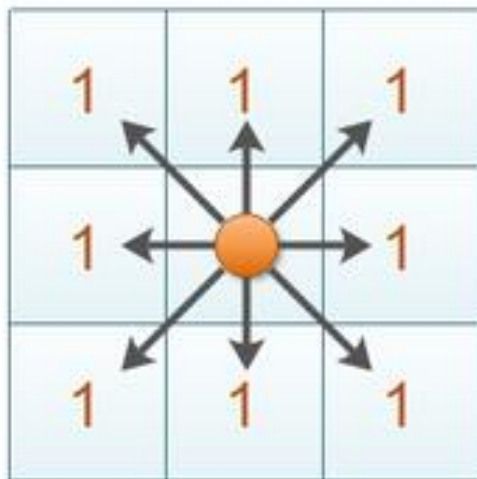
$$D_n = |X_2 - X_1| + |Y_2 - Y_1|$$

Consider a n – *dimensional* space. The Manhattan distance $d_1(p, q)$ is given by


$$d_1(p, q) = \sum_{i=1}^n |p_i - q_i|$$

What is Manhattan distance?

Chebyshev Distance



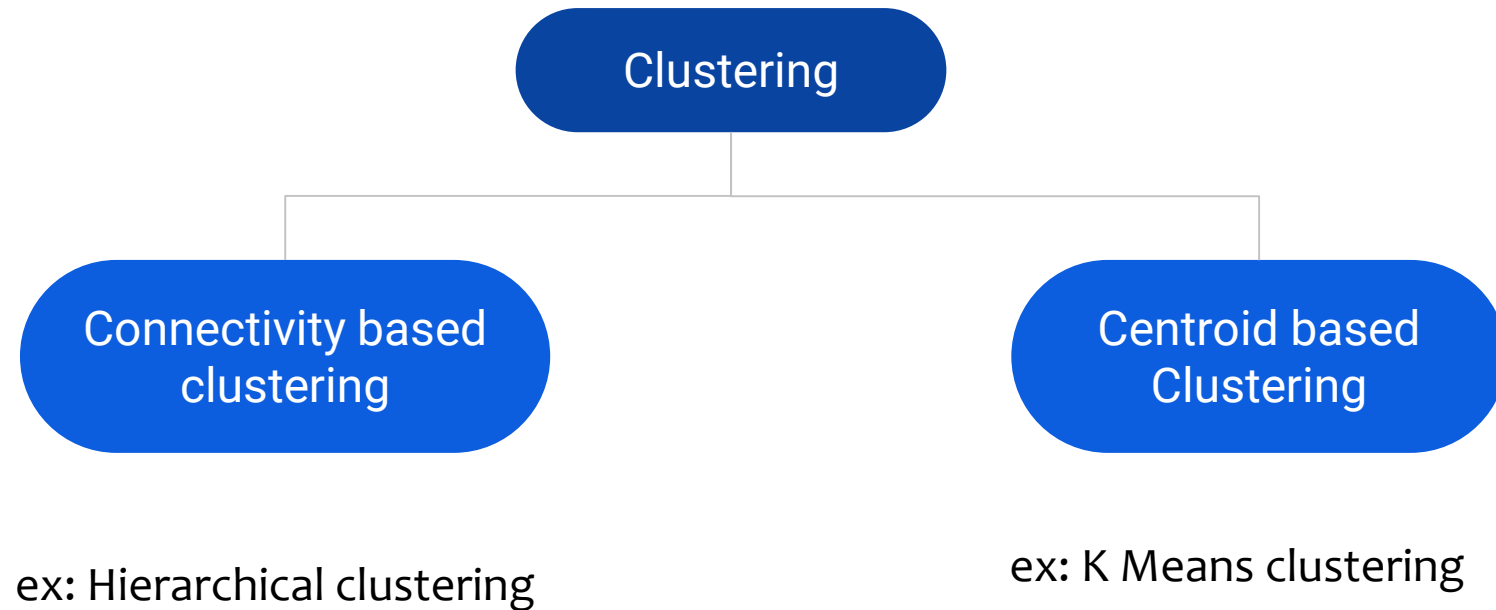
$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

The same can be extended for multiple dimensions

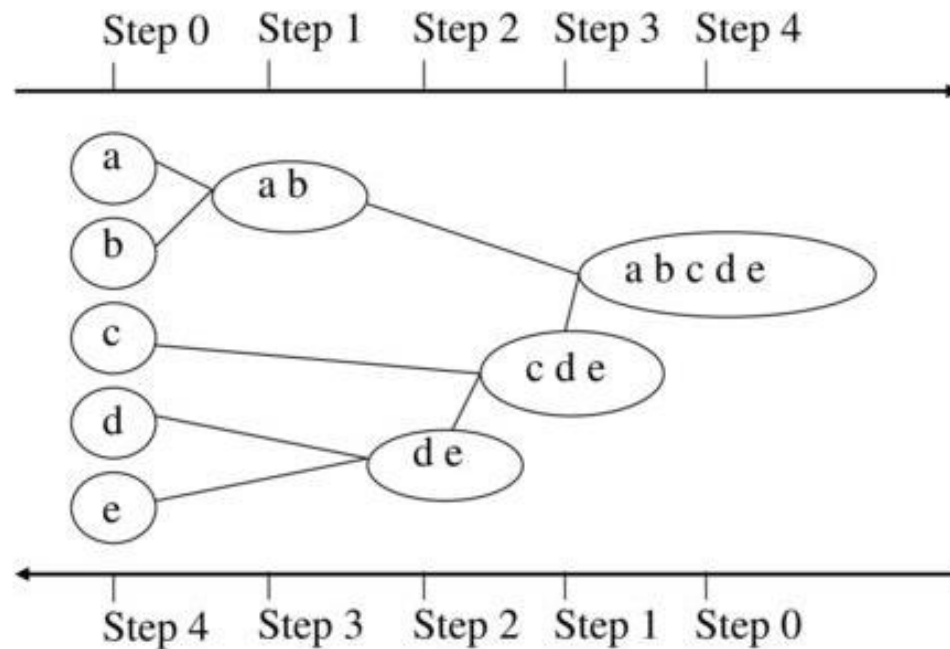
Types of clustering

greatlearning



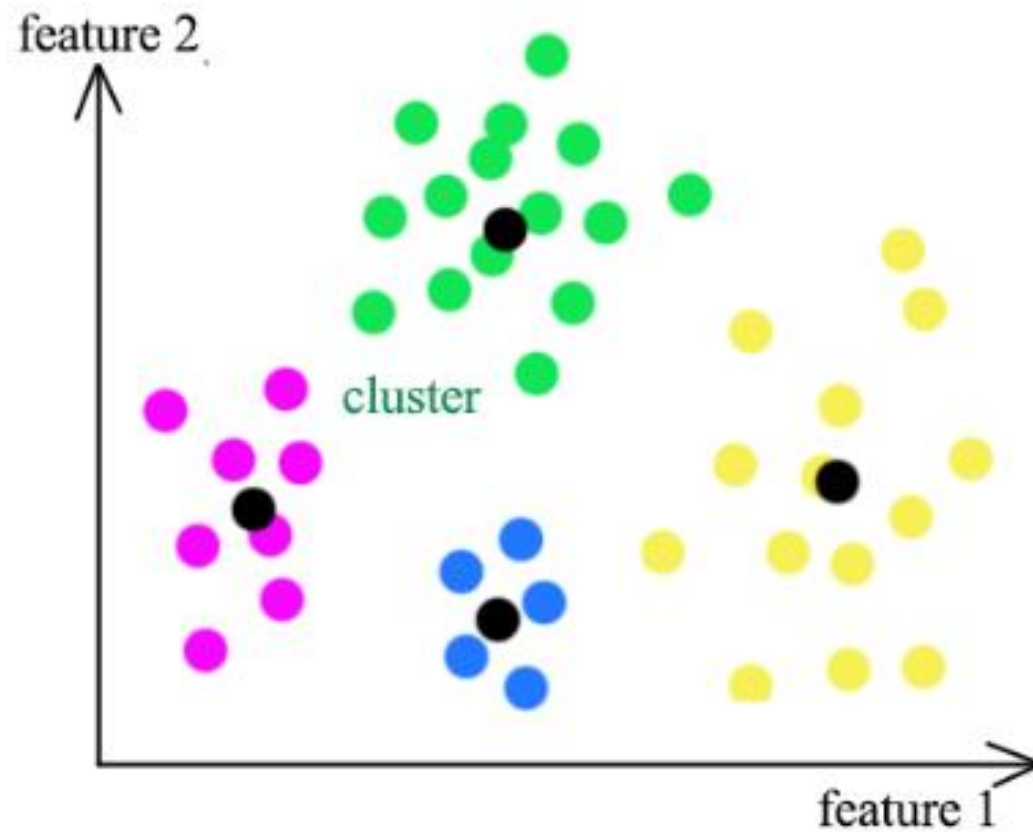
Connectivity based Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition



Computational complexity: $n(n-1)/2$ where n denoted no of data points

Centroid based Clustering

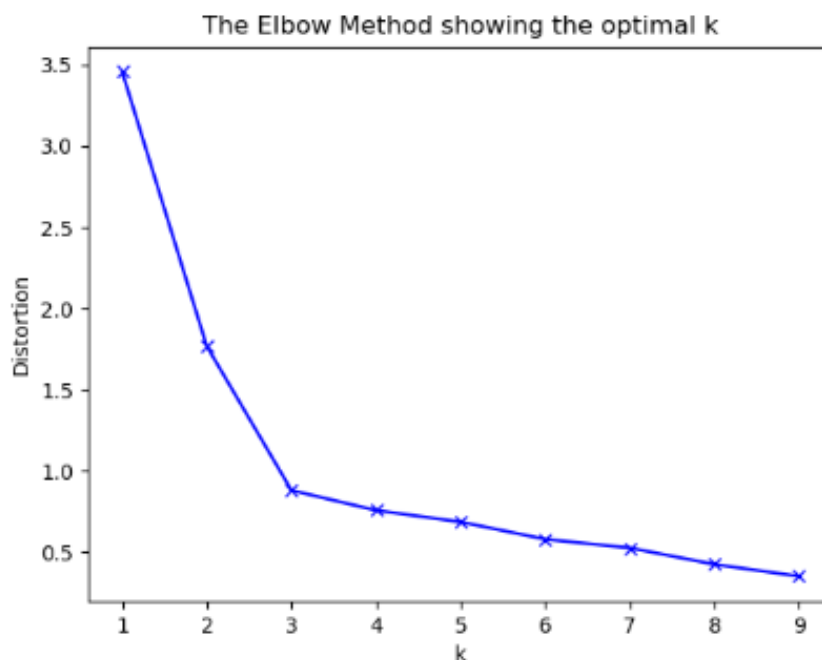


K-means clustering

- K-means clusters data by separating data points into groups of equal variance.
- It requires the number of clusters to be specified, hence the term “K” in its name
- It divides the samples into K disjoint clusters.
- The K-means algorithm chooses centroids that minimize the inertia across all the clusters.

Elbow method

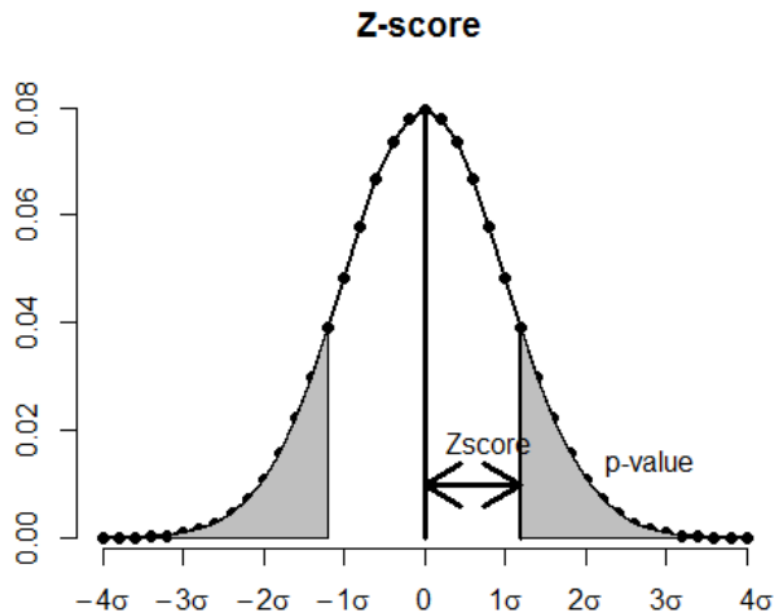
The elbow method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters.



Scaling - Z score

It is important to normalize the data using either Z score or standard scaler before performing K means clustering

This ensure different attributes are of same standard values



$$Z = \frac{x - \mu}{\sigma}$$

Score (pointing to x)

Mean (pointing to μ)

SD (pointing to σ)

More distance measures

Mahalanobis distance

$$D^2 = (\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m})$$

where:

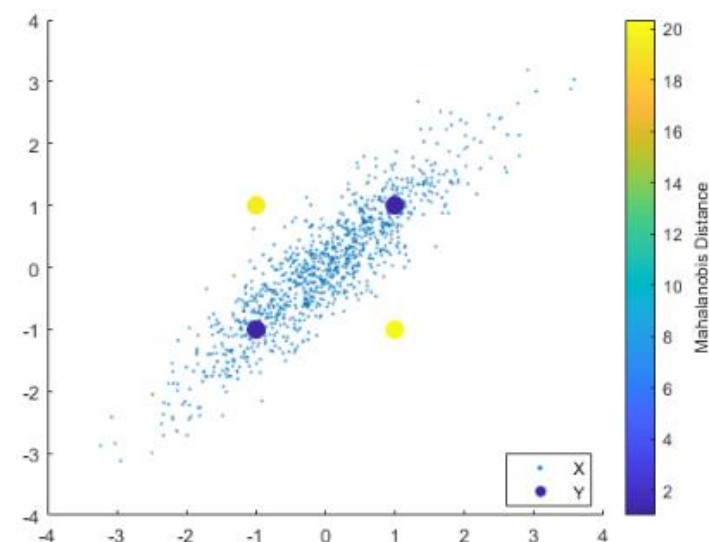
D^2 = Mahalanobis distance

\mathbf{x} = Vector of data

\mathbf{m} = Vector of mean values of independent variables

\mathbf{C}^{-1} = Inverse Covariance matrix of independent variables

\mathbf{T} = Indicates vector should be transposed



Jaccard distance

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Applications of clustering

- Image processing : used to cluster of pixels representing objects in each frame.
- Medical: Patient attributes such as age, height, weight, systolic, etc. can identify naturally occurring clusters under various health conditions.
- Customer segmentation: cluster customers on basis of frequency look for common attributes among high value customers.

Industry applications - clustering **greatlearning**

- Customer segmentation – buying patterns, income, spending behaviour, loyalty, customer lifetime value
- Anomaly detection
- Creating newsfeeds – cluster articles based on their similarity
- Pattern detection in medical imaging for diagnostics

K means clustering

- Simplest unsupervised learning algorithm
- Classify a data set through a number of clusters fixed apriori.
- The idea is to define k centroids, one for each cluster.
- The centroids should be carefully placed, as far as possible from each other.
- Each point is associated to a centroid and then centroids are re-calculated to minimize MSE.
- Uses Euclidean distance

Advantages:

- With large variables, k means is faster than other clustering algorithms.
- Produces tighter clusters than hierarchical clustering.
- Less impacted by outliers

Disadvantages:

- Predicting the number of clusters is a tedious task.
- Initial seeds have a strong impact on the final results
- If the data has clusters (in the original data) of different size and different density, the algorithm does not work²⁰ well.

Silhouette coefficient

Silhouette coefficient is a measure of how similar a datapoint is to its own cluster compared to that of other clusters. It lies in the range of $[-1,1]$

+1 = The data point is far away from the neighboring cluster and close to its own

-1 = The data point is close to other neighbouring cluster than its own cluster

0 = The data point is at the boundary of the distance between the own and neighbouring cluster

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases}$$

Visual analysis for clustering

Visual analysis for clustering

1. Visual analysis of the attributes selected for the clustering may give an idea of the range of values that K should be evaluated in.
1. Identifying the attributes on which clusters are clearly demarcated and using them in incremental order to build the multi-dimensional clusters likely to give much better clusters than using all the attributes at one go.

Dynamic clustering

- Clustering on correct attributes is the key to good clustering results.
- We can also consider those attributes whose value changes with time. For e.g. Age, income category, years of work experience etc.
- We can use sequential k means clustering over time to track individual clusters.
- Cluster size, new entries and exits one can analyse the impact of strategies designed based on earlier clustering analysis.

Hands on exercise on K-means clustering

Technical support data can often be a rich source of information on opportunities for improving customer experience. Let us analyse the tech support data and do some basic analysis on problem types, time to resolve the problem and channel of support that is most suitable.

Some important functions:

1. Importing libraries

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.cluster import KMeans
```

```
from scipy.stats import zscore
```


2. Group Data into similar clusters

for k in clusters:

```
model=KMeans(n_clusters=k)
```

```
model.fit(techSuppScaled)
```

```
prediction=model.predict(techSuppScaled)
```

```
meanDistortions.append(sum(np.min(cdist(techSuppScaled, model.cluster_centers_, 'euclidean'), axis=1)) /  
techSuppScaled.shape[0])
```

Let us first start with $K = 3$

```
final_model=KMeans(3)
```

```
final_model.fit(techSuppScaled)
```

```
prediction=final_model.predict(techSuppScaled)
```

3. Analyze the distribution of the data among the two groups ($K = 3$). One of the most informative visual tool is boxplot.

```
#plt.cla()
```

```
techSuppClust = tech_supp_df.groupby(['GROUP'])
```

```
techSuppClust.mean()
```

```
techSuppScaled.boxplot(by='GROUP', layout = (2,4),figsize=(15,10))
```

Case Study

Objective

To determine if there is a relationship between higher levels of black and white thinking and higher levels of self-reported depression in psychiatric patients hospitalized for depression. Also apply K means clustering and assign groups for model prediction.

Context:

It is common for people who tend to think of their reality as a series of black and white events to suffer from depression.

For eg;," If your child isn't "brilliant" then he must be 'stupid.' If you're not 'fascinating' then you must be 'boring.'

Steps to follow:

- Import all the necessary libraries
- Get the data
- Print the summary
- Find optimal number of clusters
- Create a new dataframe only for labels and convert it into categorical variable.
- Convert the groupdataframe created by groupby back to dataframe.
- Append the prediction with K=2
- Make scatter plots, and based on the plot make inferences on group 0 and group 1



Questions?

