SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE NAME: Machine Learning Laboratory **COURSE CODE:** DJS22L602
CLASS: Third Year B. Tech **SEM:** VI
Name: Anish Sharma Div:IT-1-1 Roll no:I011

EXPERIMENT NO. 5

CO Measured:

CO3 Apply various machine learning techniques

TITLE: To perform clustering for grouping together data points to clusters with similar characteristics and interpret results.

AIM / OBJECTIVE:
Perform cluster analysis using following methods for the selected dataset and compare results.

- K-means clustering
- K-Mediods clustering
- Agglomerative clustering
- Hierarchical clustering
- DBSCAN clustering

DESCRIPTION OF EXPERIMENT:

Clustering is basically defined as division of data into groups of similar objects. Each group called a cluster consists of objects that are similar between themselves and dissimilar compared of other groups.
Let's compare among different type of clusters.
The algorithms under discuss are: **k-means algorithm, hierarchical clustering algorithm, selforganizing maps algorithm, and expectation maximization clustering algorithm.**

Comparison Metrics:

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

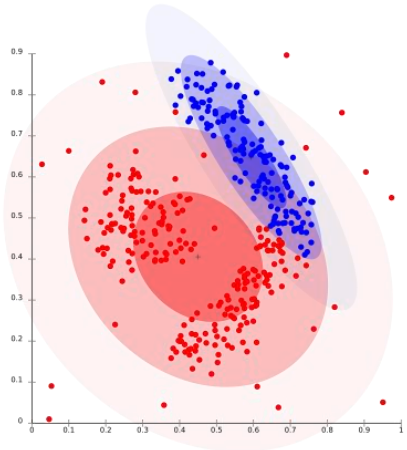
Now I would like to decide the factors on which I would discuss the comparison amongst the clustering algorithms:

1. size of dataset
2. number of clusters
3. type of dataset and type of software used
4. performance of the algorithm
5. accuracy of the algorithm
6. quality of the algorithm

How are algorithms implemented?

The clustering Algorithms are of many types. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized. **Distribution based methods:**

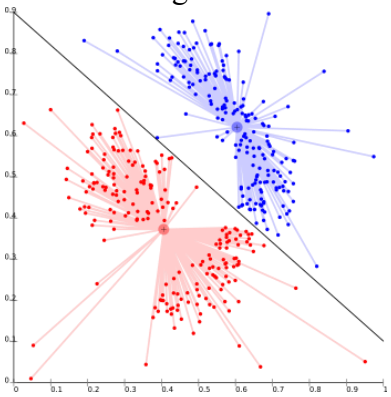
It is a clustering model in which we will fit the data on the probability that how it may belong to the same distribution. The grouping done may be *normal or gaussian*. Gaussian distribution is more prominent where we have a fixed number of distributions and all the upcoming data is fitted into it such that the distribution of data may get maximized. This result in grouping which is shown in the figure:-



SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

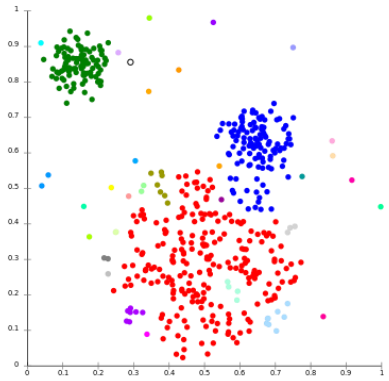
This model works well on synthetic data and diversely sized clusters. But this model may have problems if the constraints are not used to limit the model's complexity. Furthermore, Distributionbased clustering produces clusters that assume concisely defined mathematical models underlying the data, a rather strong assumption for some data distributions.
For Ex- The expectation-maximization *algorithm* which uses multivariate normal distributions is one of the popular examples of this algorithm.

Centroid based methods:
This is basically one of the iterative clustering algorithms in which the clusters are formed by the closeness of data points to the *centroid* of clusters. Here, the cluster center i.e. *centroid* is formed such that the distance of data points is minimum with the center. This problem is basically one of the NPHard problems and thus solutions are commonly approximated over a number of trials. For Ex- K – means algorithm is one of the popular examples of this algorithm.



The biggest problem with this algorithm is that we need to specify K in advance. It also has problems in clustering density-based distributions.

Connectivity based methods :
The core idea of the connectivity-based model is similar to Centroid based model which is basically defining clusters on the basis of the closeness of data points. Here we work on a notion that the data points which are closer have similar behavior as compared to data points that are farther. It is not a single partitioning of the data set, instead, it provides an extensive hierarchy of clusters that merge with each other at certain distances. Here the choice of distance function is subjective. These models are very easy to interpret but it lacks scalability.

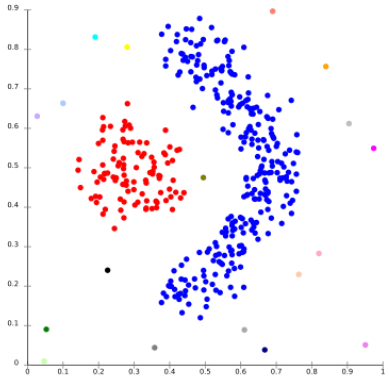


For Ex- hierarchical algorithm and its variants.

Density Models:

In this clustering model, there will be searching of data space for areas of the varied density of data points in the data space. It isolates various density regions based on different densities present in the data space.

For Ex- DBSCAN and OPTICS.



Subspace clustering :

Subspace clustering is an unsupervised learning problem that aims at grouping data points into multiple clusters so that data points at a single cluster lie approximately on a low-dimensional linear subspace. Subspace clustering is an extension of feature selection just as with feature selection subspace clustering requires a search method and evaluation criteria but in addition subspace clustering limit the scope of evaluation criteria. The subspace clustering algorithm localizes the search

for relevant dimensions and allows them to find the cluster that exists in multiple overlapping subspaces. Subspace clustering was originally purposed to solved very specific computer vision problems having a union of subspace structure in the data but it gains increasing attention in the statistic and machine learning community. People use this tool in social networks, movie recommendations, and biological datasets. Subspace clustering raises the concern of data privacy as many such applications involve dealing with sensitive information. Data points are assumed to be incoherent as it only protects the differential privacy of any feature of a user rather than the entire profile user of the database.
There are two branches of subspace clustering based on their search strategy.

- Top-down algorithms find an initial clustering in the full set of dimensions and evaluate the subspace of each cluster.
- The bottom-up approach finds dense region in low dimensional space then combine to form clusters.

PROCEDURE:

1 Perform cluster analysis using following methods for given case study and compare results.

1. K-means clustering
2. K-Mediods clustering
3. Agglomerative clustering
4. Hierarchical clustering
5. DBSCAN clustering

2 Compare the models w.r.t.

- number of clusters
- accuracy of the algorithm
- Time and space complexity

Dataset: Credit Card Dataset
Credit Card Dataset for Clustering | Kaggle

Help guide:
A Comparative Study of Clustering Algorithms | by ishika chatterjee | Analytics Vidhya | Medium
Comparing the performance of different machine learning algorithms - Dibyendu Deb

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

OBSERVATIONS / DISCUSSION OF RESULT:

1. Compare between below clustering algorithms and discuss about their performance.
   - K-means clustering
   - K-Mediods clustering
   - Agglomerative clustering
   - Hierarchical clustering
   - DBSCAN clustering CODE:

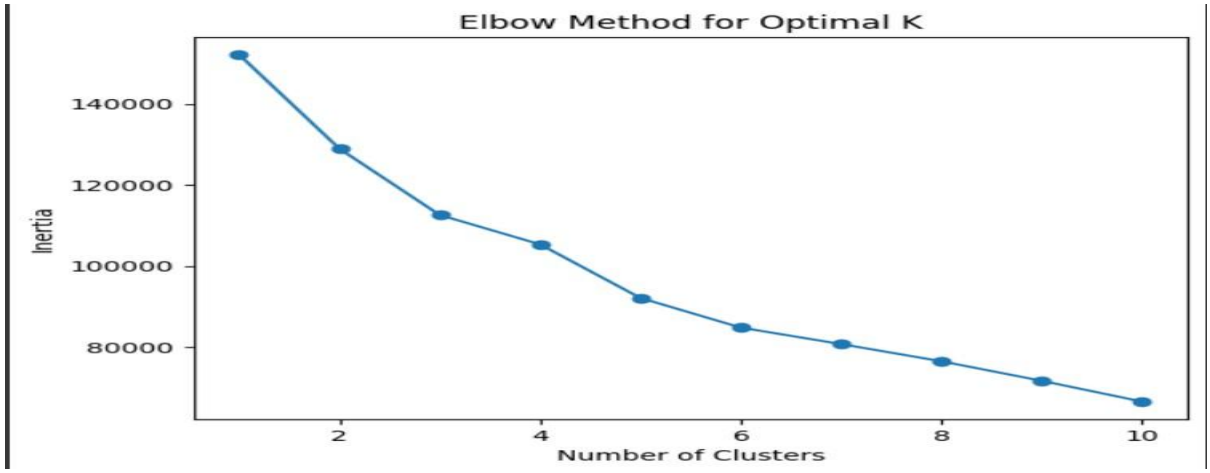
```
import pandas as pd import numpy as np import matplotlib.pyplot as plt
import seaborn as sns from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
from sklearn.metrics import silhouette_score from scipy.cluster.hierarchy
import dendrogram, linkage from sklearn_extra.cluster import KMedoids

# Load the dataset df =
pd.read_csv("CC GENERAL.csv") #
Drop unnecessary columns
df.drop(['CUST_ID'], axis=1,
inplace=True)

# Handle missing values df.fillna(df.mean(),
inplace=True)

# Standardize the data scaler
= StandardScaler()
df_scaled = scaler.fit_transform(df)

# K-Means Clustering
inertia = [] for k in
range(1, 11):
   kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(df_scaled)
   inertia.append(kmeans.inertia_)
```

```
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia') plt.title('Elbow
Method for Optimal K') plt.show()

kmeans = KMeans(n_clusters=3, random_state=42) df['KMeans_Cluster']
= kmeans.fit_predict(df_scaled)
print("Silhouette Score (K-Means):", silhouette_score(df_scaled, df['KMeans_Cluster']))

# K-Medoids Clustering kmedoids =
KMedoids(n_clusters=3, random_state=42)
df['KMedoids_Cluster'] = kmedoids.fit_predict(df_scaled)
print("Silhouette Score (K-Medoids):", silhouette_score(df_scaled, df['KMedoids_Cluster']))

# Agglomerative Clustering
agg_clust = AgglomerativeClustering(n_clusters=3, linkage='ward') df['Agglomerative_Cluster']
= agg_clust.fit_predict(df_scaled)
print("Silhouette Score (Agglomerative):", silhouette_score(df_scaled, df['Agglomerative_Cluster']))

# Hierarchical Clustering Dendrogram plt.figure(figsize=(10,
5)) linkage_matrix = linkage(df_scaled, method='ward')
dendrogram(linkage_matrix) plt.title('Dendrogram for
Hierarchical Clustering') plt.xlabel('Data Points')
plt.ylabel('Distance') plt.show()

# DBSCAN Clustering
dbscan = DBSCAN(eps=1.5, min_samples=5) df['DBSCAN_Cluster']
= dbscan.fit_predict(df_scaled)
print("Unique Clusters (DBSCAN):", np.unique(df['DBSCAN_Cluster']))

# Comparison of Clustering Algorithms models = ['K-
Means', 'K-Medoids', 'Agglomerative'] silhouette_scores =
[     silhouette_score(df_scaled, df['KMeans_Cluster']),
silhouette_score(df_scaled, df['KMedoids_Cluster']),
silhouette_score(df_scaled, df['Agglomerative_Cluster'])
]
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

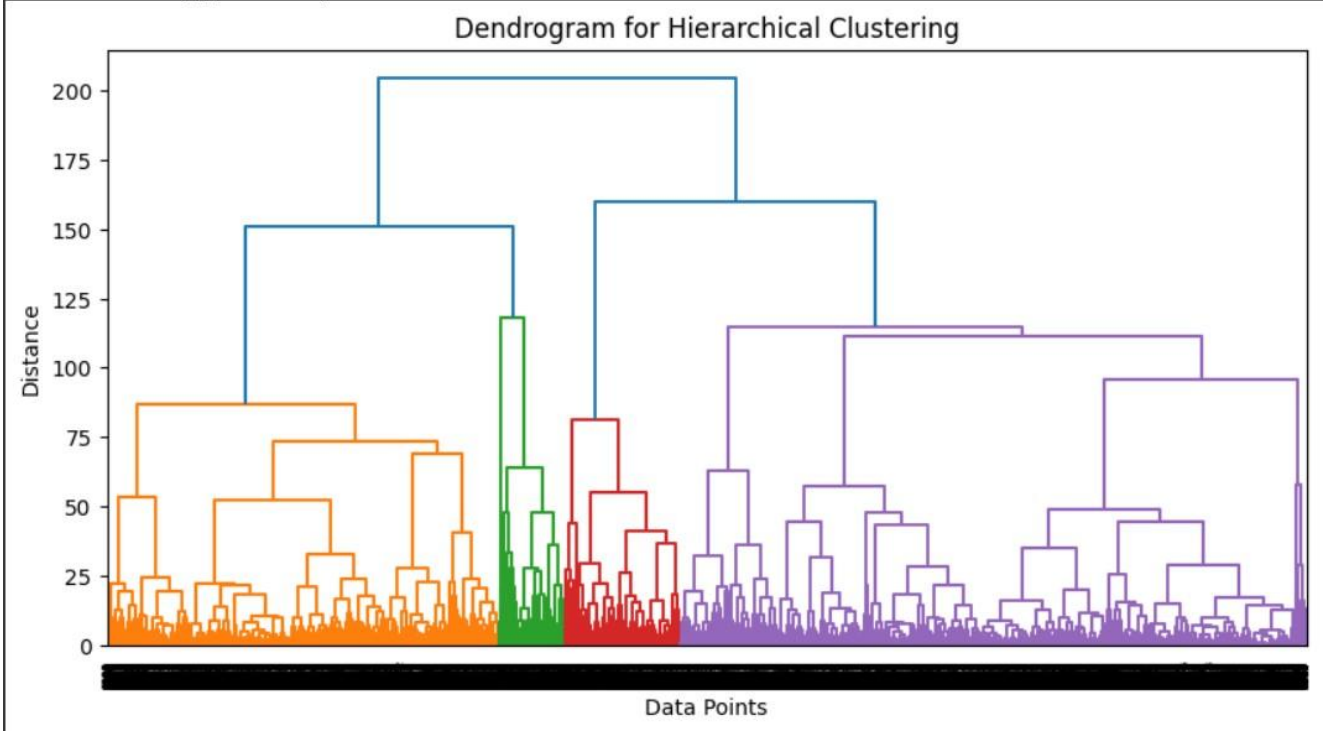
```
plt.figure(figsize=(8, 5)) plt.bar(models, silhouette_scores, color=['blue', 'green', 'red']) plt.xlabel('Clustering Algorithm') plt.ylabel('Silhouette Score') plt.title('Comparison of Clustering Algorithms') plt.show()
```

OUTPUT:

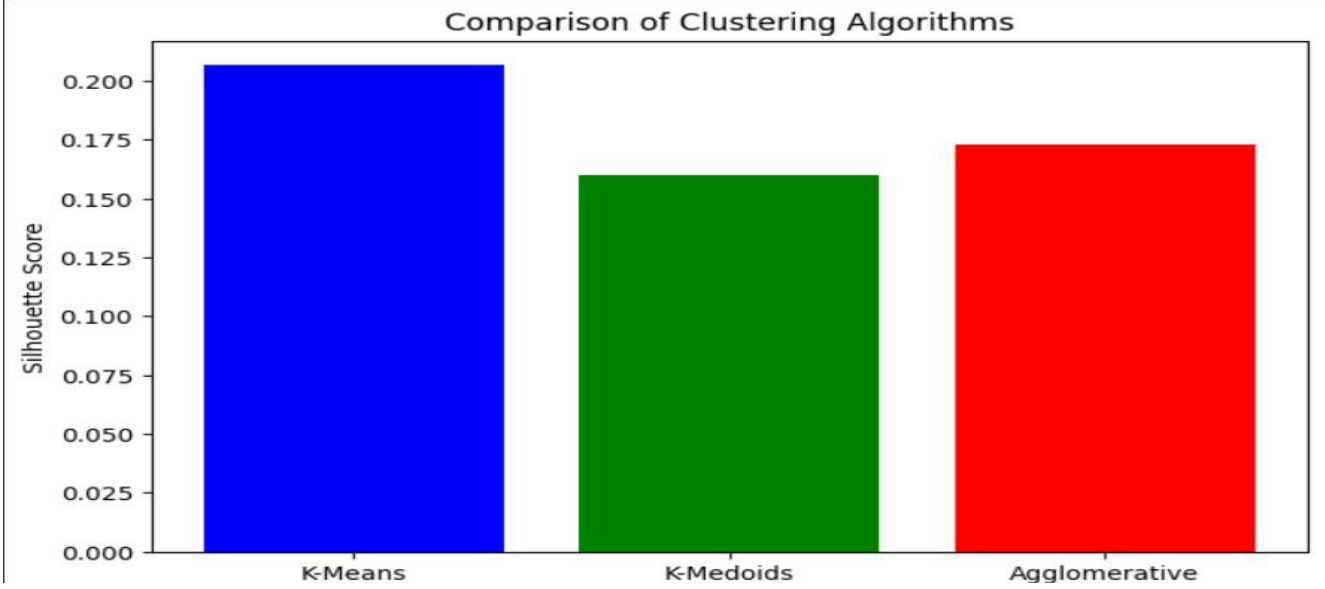


SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
Silhouette Score (K-Means): 0.20676101192444302
Silhouette Score (K-Medoids): 0.1601239148718294
Silhouette Score (Agglomerative): 0.1731098007232828
```




```
Unique Clusters (DBSCAN): [-1  0  1  2  3  4  5]
```



SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

CONCLUSION:

- Hierarchical Clustering provides a clear structure using a dendrogram but is computationally expensive for large datasets.
- DBSCAN automatically detects clusters and outliers but is sensitive to parameter selection (eps and min_samples).
- Comparison: Hierarchical is good for small datasets with clear relationships, while DBSCAN is better for large, noisy datasets with irregular cluster shapes. • Final Takeaway: The best algorithm depends on dataset size, noise, and cluster distribution.

REFERENCES:

(List the references as per format given below and citations to be included the document)

1. Ethem Alpaydın, “Introduction to Machine Learning”, 4th Edition, The MIT Press, 2020.
2. Peter Harrington, “Machine Learning in Action”, 1st Edition, Dreamtech Press, 2012.
3. Tom Mitchell, “Machine Learning”, 1st Edition, McGraw Hill, 2017.
4. Andreas C, Müller and Sarah Guido, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, 1st Edition, O'reilly, 2016. 5. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, 1st Edition, MIT Press, 2012.

Website References: https://developers.google.com/machine-learning/clustering/overview
https://www.geeksforgeeks.org/clustering-in-machine-learning/
https://www.sciencedirect.com/science/article/pii/S095219762200046X
https://www.sciencedirect.com/science/article/abs/pii/B9780128157398000134