Sane's STATS Academy of Statistics

Cluster Analysis

Sane's STATS Academy of Statistics

Need for Clustering

- Groups available for grouping Customers
  - Male and female
  - High Income, Medium Income, Low Income
  - Having average invoice amounts as < 1Lac, between 5 Lac and 10 Lac etc.
  - Customers making Cheque, Cash or card payments
  - Any other type of grouping
- As per our convenience we categorize the entities based on some attributes

Need for Clustering

- There may be a business need of grouping the business entities on some variables.
- Such as we may group the products not just on the profit margin but also considering the variables such as quantity sold, average shelf life, maintenance cost etc.

Why is Clustering done?

- Cluster analysis is used to form groups or clusters of similar records based on several measurements made on these records.

Sane's STATS Academy of Statistics

Clustering in Marketing

- Market Segmentation: Customers are segmented based on demographic and transaction history information and a marketing strategy is tailored for each segment.
- Market structure analysis: Identifying groups of similar products according to competitive measures of similarity.

Clustering in Finance

- Creating balanced portfolios: Given data on a variety of investment opportunities (e.g., stocks), one may find clusters based on financial performance variables such as return (daily, weekly, or monthly), volatility and other characteristics, such as industry and market capitalization.

Clustering in Pure Sciences

- Biologists have made extensive use of classes and subclasses to organize species.
- In chemistry, Mendeleeyev's periodic table of the elements.

Distance Method

- For record i we have the vector of p measurements (xi1, xi2, . . . , xip), while for record j we have the vector of measurements (xj1, xj2, . . . , xjp).
- The most popular distance measure is the Euclidean distance, dij , which between two cases, i and j, is defined by

$$d_{ij} = \sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\ldots+(x_{ip}-x_{jp})^2}$$

Other Distance Measures

- Numerical Data
  - Correlation-based similarity
  - Statistical distance (also called Mahalanobis distance)
  - Manhattan distance (“city block”)
  - Maximum coordinate distance
- Categorical Data
  - Matching coefficient: (a + d)/p
  - Jaquard’s coefficient: d/(b+c+d)

Scaling the data

- If the variables in analysis vary in range, the variables with large values will have larger impact on the results.
- To avoid this undesirable, we should scale the data
- Variable can be scaled in the following ways:
  - Subtract mean from each value(centering) and divide it by its standard deviation(scaling)
  - Divide each value in the variable by maximum value of the variable
  - Subtract mean from each value(centering) and divide it by its mean deviation about mean(scaling)

Sane's STATS Academy of Statistics

Types of Clustering Methods

- Hierarchical methods
  - **Agglomerative**
  - Divisive
- Nonhierarchical methods
  - **K-Means**
  - DBSCAN

Sane's STATS Academy of Statistics

Hierarchical Clustering

Distance Between Two Clusters

- Minimum distance (single linkage): the distance between the pair of records Ai and Bj that are closest.
- Maximum distance (complete linkage): the distance between the pair of records Ai and Bj that are farthest.
- Average distance (average linkage): the average distance of all possible distances between records in one cluster and records in the other cluster.

Distance Between Two Clusters

- Centroid distance: the distance between the two cluster centroids. A cluster centroid is the vector of measurement averages across all the records in that cluster.
- Ward: ANOVA sum of squares between the two clusters added up over the variables

Agglomerative Hierarchical Method

- Agglomerative method begins with n (No. of observations) clusters and sequentially merge similar clusters until a single cluster is left.

Algorithm

1. Start with n clusters (each observation = cluster).
2. The two closest observations are merged into one cluster.
3. At every step, the two clusters with the smallest distance are merged. This means that either single observations are added to existing clusters or two existing clusters are combined.

Hierarchical Clustering in Python

- Hierarchical Clustering in Python can be implemented using function scipy.cluster.hierarchy.linkage()

Syntax : scipy.cluster.hierarchy.linkage(y, method='single', metric='euclidean',..)

Where

y : data

method : can have values as 'single', 'complete', 'average', etc.

metric : can have values as chebyshev', 'cityblock', 'correlation', 'cosine', 'dice', 'euclidean', 'hamming', 'jaccard', 'jensenshannon', 'kulsinski', 'mahalanobis', 'matching', 'minkowski' etc.

Example: Milk

- The data set contains the ingredients of mammal's milk of 25 animals.
- A data frame with 25 observations on the following 5 variables (all in percent)
  - water
  - protein
  - fat
  - lactose
  - ash
- Here, we are interested in grouping the 25 mammals based on the above given 5 nutrient measures

Sane's STATS Academy of Statistics

Program: Dendrogram

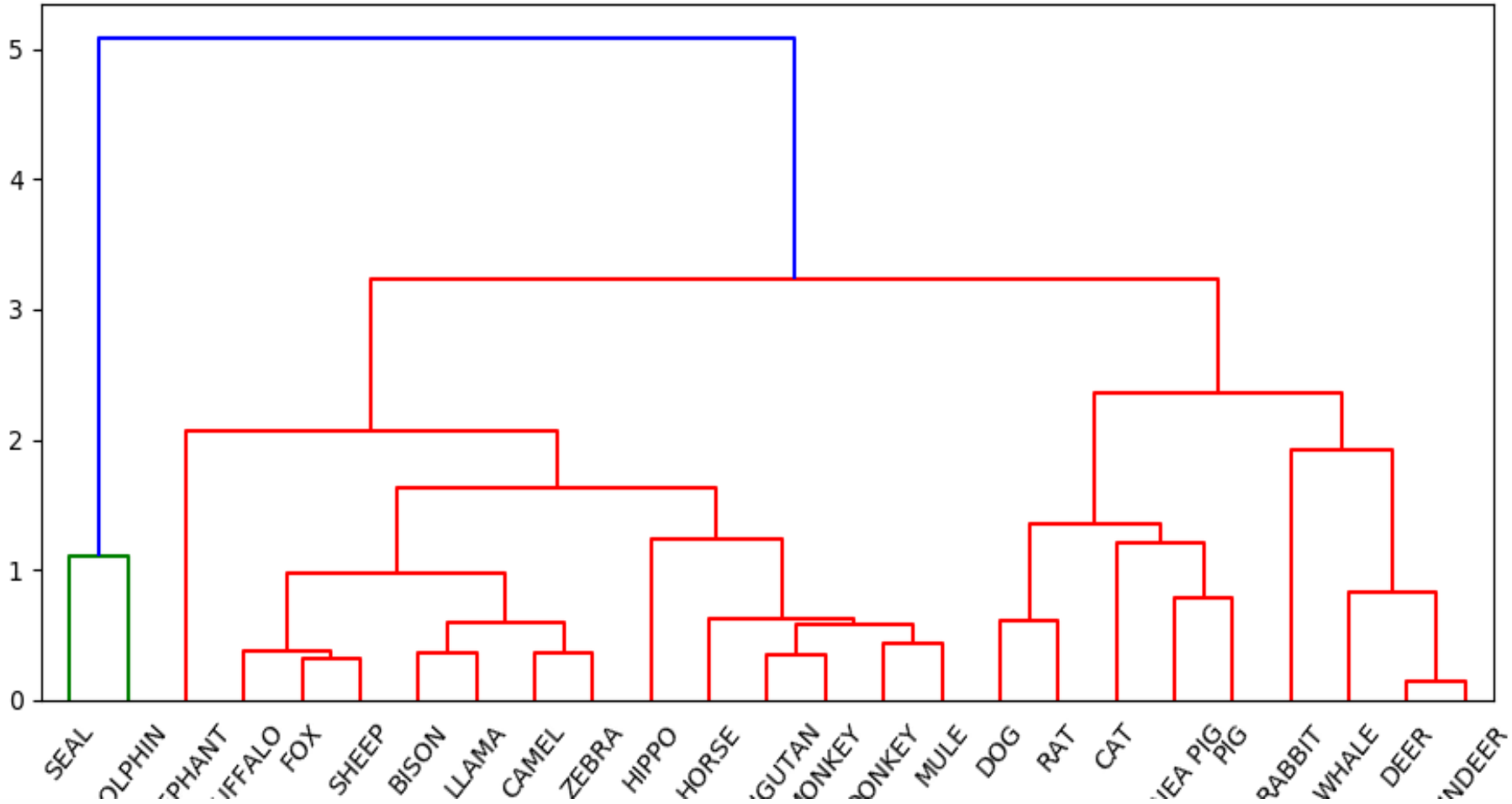
```
# Perform the necessary imports
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt

# Calculate the linkage: mergings
mergings = linkage(milkscaled,method='average')
fig_size = plt.rcParams["figure.figsize"]
fig_size[0] = 12
fig_size[1] = 8
plt.rcParams["figure.figsize"] = fig_size
# Plot the dendrogram, using varieties as labels
dendrogram(mergings,
           labels=np.array(milk.index),
           leaf_rotation=50,
           leaf_font_size=10,
)

plt.show()
```

Sane's STATS Academy of Statistics

Output: Dendrogram

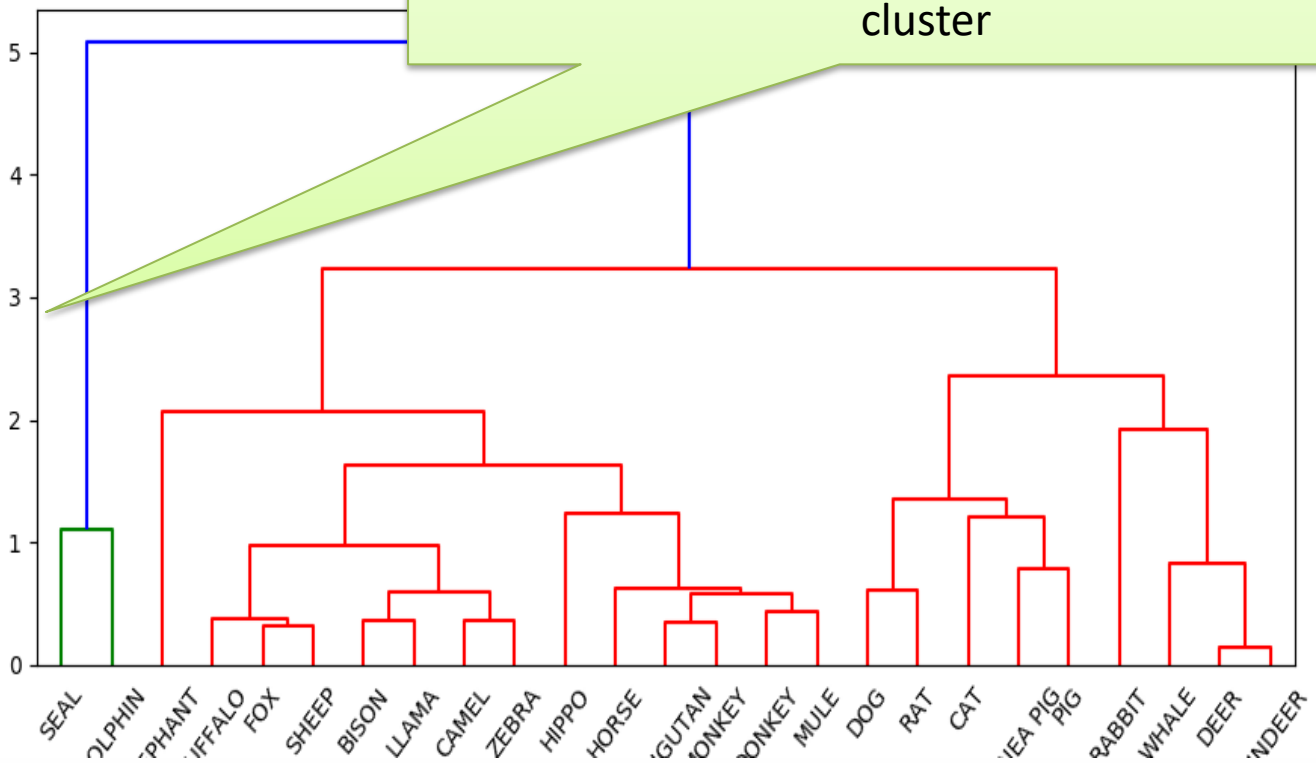


Dendrogram

- Dendrogram displays how items are combined into clusters and is read from the bottom up.
- Each observation starts as its own cluster.
- Two the observations that are closest are combined
- Then the clustered observations are combined with the closest further
- This continues until all observations are combined into a single cluster

Dendrogram

Height indicates the criterion value at which clusters are joined. For average linking, this criterion is the average distance between each point in one cluster and each point in another cluster



Limitations of Hierarchical Clustering

- Hierarchical clustering requires the computation and storage of an n × n distance matrix. For very large datasets, this can be expensive and slow.
- The hierarchical algorithm makes only one pass through the data. This means that records that are allocated incorrectly early in the process cannot be reallocated subsequently.

Limitations of Hierarchical Clustering

- With respect to the choice of distance between clusters, single and complete linkage are robust to changes in the distance metric (e.g., Euclidean, statistical distance) as long as the relative ordering is kept.
- Hierarchical clustering is sensitive to outliers.

Sane's STATS Academy of Statistics

K-Means Clustering

K-Means

- Forming good clusters is to pre-specify a desired number of clusters, k, and to assign each case to one of k clusters so as to minimize a measure of dispersion (variation) within the clusters.
- The method divides the sample into a predetermined number k of non-overlapping clusters so that clusters are as homogeneous as possible with respect to the measurements used.

Algorithm

1. Start with k initial clusters (user chooses k). The starting points are chosen by software at random
2. At every step, each record is reassigned to the cluster with the “closest” centroid.
3. Re-compute the centroids of clusters that lost or gained a record, and repeat step 2.
4. Stop when moving any more records between clusters increases cluster dispersion.

Sane's STATS Academy of Statistics

K-means in Python

Syntax : sklearn.cluster.kmeans(X, n_clusters, max_iter=300, random_state=None, …)

Where

X: array-like or sparse matrix, shape

n_clusters : number of clusters

max_iter : Maximum number of iterations of the k-means algorithm to run

random_state : int, RandomState instance or None (default)

Sane's STATS Academy of Statistics

Program & Output

```
In [29]: from sklearn.preprocessing import StandardScaler
    ...: # Create scaler: scaler
    ...: scaler = StandardScaler()
    ...: milkscaled=scaler.fit_transform(milk)
    ...:
    ...: # Import KMeans
    ...: from sklearn.cluster import KMeans
    ...:
    ...: # Create a KMeans instance with clusters: model
    ...: model = KMeans(n_clusters=3)
    ...:
    ...: # Fit model to points
    ...: model.fit(milkscaled)
    ...: #model.n_init
    ...: # Determine the cluster labels of new_points: labels
    ...: labels = model.predict(milkscaled)
    ...:
    ...: # Print cluster labels of new_points
    ...: print(labels)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 2 1 1]
```

Sane's STATS Academy of Statistics

Within Sum of Squares

- Within Sum of Squares indicate the variation
- We can create a function to extract and plot it.
- Lower the WSS better is the cluster variation.
- In python, .inertia_ it can be extracted

Sane's STATS Academy of Statistics

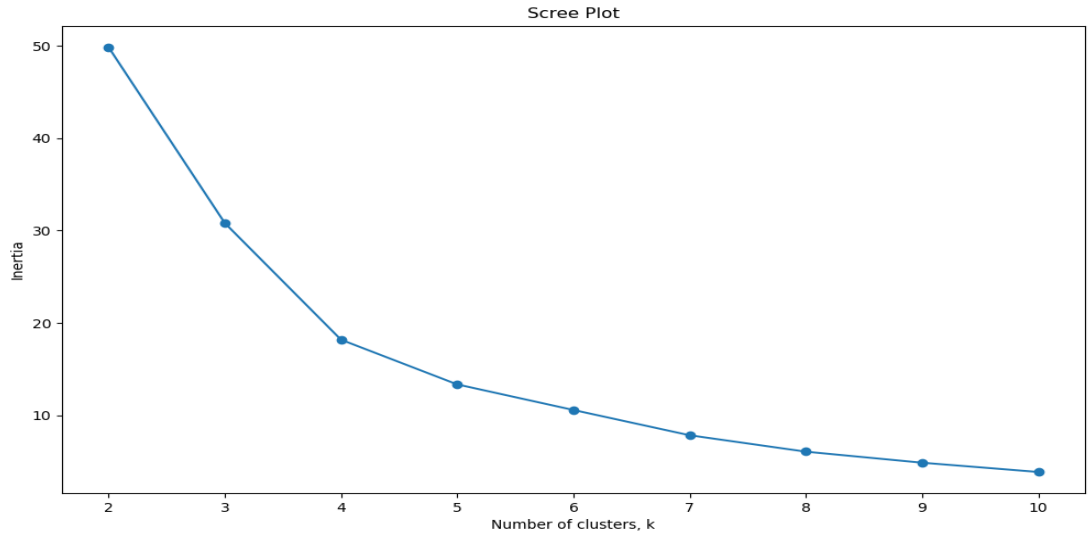
Program & Output

```
clustNos = [2,3,4,5,6,7,8,9,10]
Inertia = []

for i in clustNos :
    model = KMeans(n_clusters=i)
    model.fit(milkscaled)
    Inertia.append(model.inertia_)

# Import pyplot
import matplotlib.pyplot as plt

plt.plot(clustNos, Inertia, '-o')
plt.title("Scree Plot")
plt.xlabel('Number of clusters, k')
plt.ylabel('Inertia')
plt.xticks(clustNos)
plt.show()
```



Silhouette Score

- The Silhouette Score is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each observation.
- The Silhouette Coefficient for any observations is (b - a) / max(a, b).
- In other words, b is the distance between that observation and the nearest cluster that the observation is not a part of.
- **Silhouette score** is calculated as mean of all the silhouette coefficients for all the observations
- Note that Silhouette Coefficient is only defined if number of labels is 2 <= n_labels <= n_observations – 1
- Bigger the Score better is the clustering

Good Explanation: https://www.youtube.com/watch?v=_jg1UFoef1c&t=140s

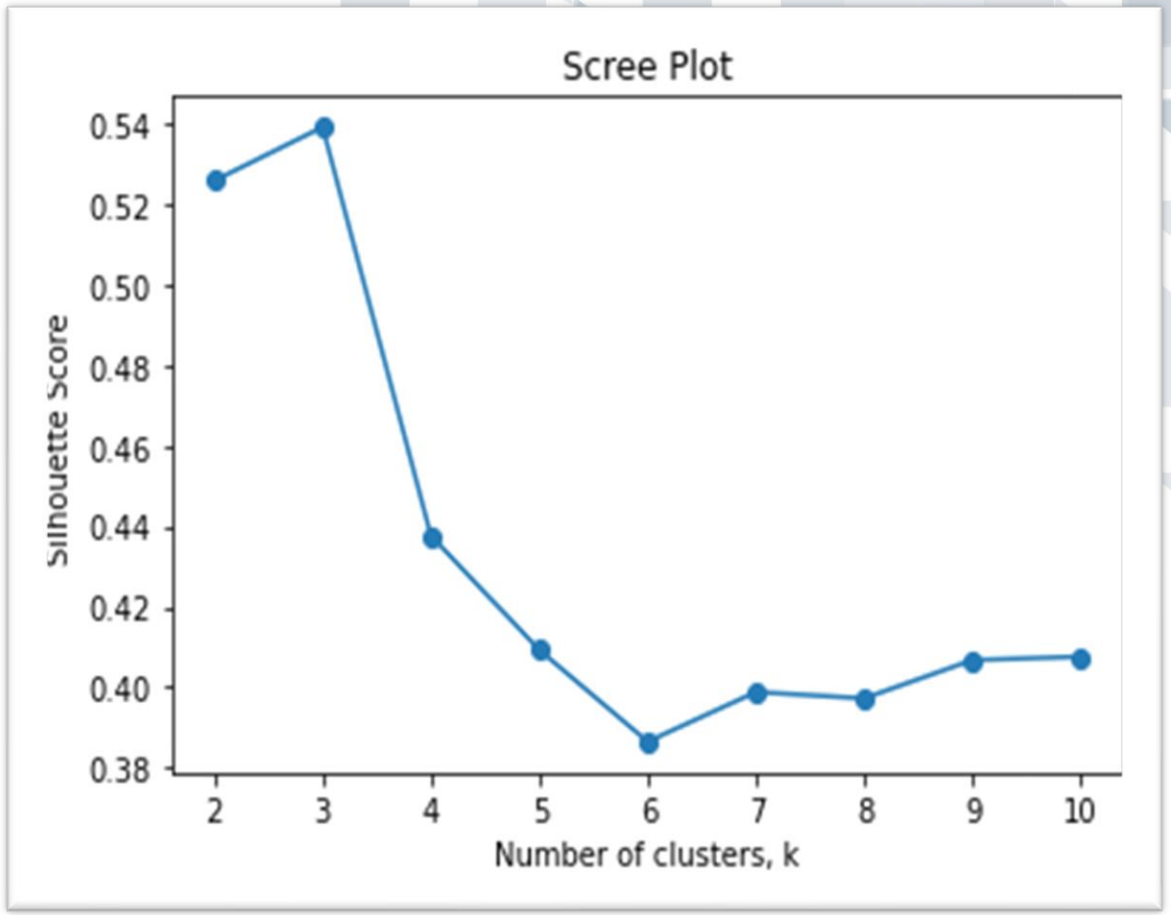
Program and Output

```
clustNos = [2,3,4,5,6,7,8,9,10]
silhouettes = []

for i in clustNos :
    model = KMeans(n_clusters=i,random_state=2021)
    model.fit(milkscaled)
    labels = model.predict(milkscaled)
    sil_score = silhouette_score(milkscaled,labels)
    silhouettes.append(sil_score)

# Import pyplot
import matplotlib.pyplot as plt

plt.plot(clustNos, silhouettes, '-o')
plt.title("Scree Plot")
plt.xlabel('Number of clusters, k')
plt.ylabel('Silhouette Score')
plt.xticks(clustNos)
plt.show()
```



Sane's STATS Academy of Statistics

Advantages of K-Means Clustering

- K-means clustering can handle larger datasets than hierarchical cluster approach.
- Observations are not permanently committed to any cluster but, they are changed at every iteration.

Sane's STATS Academy of Statistics

Limitations

- All the variables have to be continuous / numeric
- Clusters are severely affected by outliers
- Clusters are sensitive to initialization
- Clusters obtained are of differing densities