**Hierarchical Clustering**

**Instructions:**

Please share your answers wherever applicable in line with the Word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: Vishvash C Batch ID: 23012024**

**Topic: Hierarchical Clustering**

**Guidelines:**

**1. An assignment submission is considered complete only when correct and executable code(s) is submitted along with the documentation, explaining the method and results. Failing to submit either of those will be considered an invalid submission and will not be considered a correct submission.**

**2. Ensure that you submit your assignments correctly and in full. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

**Hints:**

**1. Business Problem**

* 1. **What is the business objective?**
  2. **What are the constraints?**
  3. **Define success criteria**

**2. Work on each feature of the dataset to create a data dictionary as displayed in the below image:**



**3. Exploratory Data Analysis (EDA):**

**3.1. Univariate analysis.**

**3.2. Bivariate analysis.**

**4. Data Pre-processing**

**4.1 Data Cleaning, Feature Engineering, etc.**

**5. Model Building**

**5.1 Build the model on the scaled data (try multiple options).**

**5.2 Perform the hierarchical clustering and visualize the clusters using a dendrogram.**

**5.3 Validate the clusters (try with the different numbers of clusters), label the clusters, and derive insights (compare the results from multiple approaches).**

**6. Write about the benefits/impact of the solution - in what way does the business (client) benefit from the solution provided?**

**Problem Statements:**

Global air travel has seen an upward trend in recent times. The maintenance of operational efficiency and maximizing profitability are crucial for airlines and airport authorities. Businesses need to optimize airline and terminal operations to enhance passenger satisfaction, improve turnover rates, and increase overall revenue.

The airline companies with the available data want to find an opportunity to analyze and understand travel patterns, customer demand, and terminal usage.

**CRISP-ML(Q) process model describes six phases:**

1. Business and Data Understanding

2. Data Preparation

3. Model Building

4. Model Evaluation

5. Deployment

6. Monitoring and Maintenance

**Objective**: Maximize the operational efficiency

**Constraints**: Maximize the financial health

**Success Criteria:**

Business Success Criteria: Increase the operational efficiency by 10% to 12% by segmenting the Airlines.

ML Success Criteria: Achieve a Silhouette coefficient of at least 0.7

Economic Success Criteria: The airline companies will see an increase in revenues by at least 8% (hypothetical numbers)

Data: Refer to the ‘AirTraffic\_Passenger\_Statistics.csv’ dataset.

**Questions to Trigger Your Thoughts:**

**Answers:**

Q1. Why it is important to define the objectives for any Business problem?

Defining objectives helps in understanding the problem's scope and goals, guiding the development of strategies and solutions, and measuring the success of the project.

Q2. How to maintain the quality of the Machine Learning model developed for the Business problem?

Quality maintenance involves regular monitoring of model performance, updating data and algorithms as needed, conducting thorough testing, and implementing feedback loops for continuous improvement.

Q3. What is the first document created/drafted for any ML project?

The first document is typically a project charter, outlining the project objectives, scope, resources, timelines, and deliverables.

Q4. How to load data with multiple sheets?

In Python, you can use libraries like pandas to load data with multiple sheets from Excel files using the pd.read\_excel() function with the sheet\_name parameter set to either a list of sheet names or None to load all sheets.

Q5. What are the Auto EDA techniques?

Auto EDA (Automated Exploratory Data Analysis) techniques include libraries and tools like pandas-profiling, AutoViz, and Sweetviz, which automate the process of data exploration, visualization, and summary statistics generation.

Q6. What are four business moments, and what insights we can draw from them?

Measure of central tendency: Mean: Represents the average value of a dataset, providing a central tendency measure.

Measure of dispersion: Variance: Indicates the spread of data points around the mean, highlighting the dispersion of the dataset.

Measure of asymmetry: Skewness: Reflects the asymmetry of the data distribution, showing whether it is skewed to the left or right.

Measure of peakedness: Kurtosis: Describes the peakedness or flatness of a distribution, indicating how data is distributed around the mean

Q7. Write the techniques in data Pre-Processing.

Techniques in data pre-processing include data cleaning (handling missing values, outliers), data transformation (scaling, encoding categorical variables), and feature engineering (creating new features, dimensionality reduction).

Q8. When we use label encoding and one-hot encoding?

Label encoding is used for converting categorical variables into numerical labels when the categories have an ordinal relationship. One-hot encoding is used when categories are nominal and don't have an ordinal relationship.

Q9. What is the technique to remove outliers?

Techniques to remove outliers include statistical methods like Z-score, IQR (Interquartile Range), winzorisation, trimming, etc.,

Q10. What are the techniques to check whether the data is normally distributed or not?

Techniques to check for normality include visual inspection using histograms, Q-Q plots, and descriptive statistics like skewness and kurtosis.

Q12. How to make data scale-free?

Data can be made scale-free by using techniques like normalization (scaling features to a specified range between 0 and 1) or standardization (scaling features to have mean 0 and standard deviation 1).

Q13. What types of graphs are used to depict the bivariate analysis?

Scatter plots, line graphs are commonly used to depict bivariate analysis, showing the relationship between two variables.

Q14. What do you mean by bivariate frequency distribution?

Bivariate frequency distribution shows the frequency or count of occurrences of combinations of two variables in a dataset, often presented in a two-way table or heatmap.

Q15. Which libraries are used in Hierarchical clustering?

Python libraries like scipy and scikit-learn are commonly used for hierarchical clustering algorithms, such as AgglomerativeClustering.

Q16. What is the difference between Agglomerative clustering and Divisive Clustering?

Agglomerative clustering starts with individual data points as clusters and iteratively merges them into larger clusters, while Divisive clustering starts with one cluster containing all data points and iteratively splits it into smaller clusters.

Q17. Which metric is used to find distance/similarities between two data points and between a record and a cluster?

Common metrics used for distance/similarity measurement include Euclidean distance, Manhattan distance

Distance metrics between clusters - record & cluster, cluster & cluster

Single Linkage - Nearest Neighbour : Minimum distance between the members of the two clusters (chaining effect)

Complete Linkage - Farthest Neighbour : Maximum distance between the members of the two clusters (may be influenced by outliers)

Average Linkage - Average of all distances between the members of the two clusters

Centroid Linkage - Distance between the centroid of two clusters

Ward's Linkage - Error sum of squares should be 0 or minimum. Less loss of information

Q18. What are the parameters needed to plot the Dendrogram?

The parameters needed to plot a dendrogram include the linkage method (e.g., complete, single, average), the distance metric, and the labels for the data points.

Q19. How to perform cluster evaluation? Which are the techniques used for cluster evaluation?

Cluster evaluation involves techniques like silhouette analysis, Davies-Bouldin Index, and Calinski-Harabasz Index, which measure the quality of clustering results based on intra-cluster cohesion and inter-cluster separation.

Q20. What do the Silhouette coefficient, Calinski Harbaz, and Davies-Bouldin Index indicate in hierarchical clustering?

These metrics indicate the quality of clustering results: Silhouette coefficient measures how well-separated clusters are, Calinski-Harabasz Index measures the ratio of between-cluster dispersion to within-cluster dispersion, and Davies-Bouldin Index measures the average similarity between each cluster and its most similar cluster.



**Code:**

# -\*- coding: utf-8 -\*-

"""

Created on Tue Mar 19 19:56:19 2024

@author: Lenovo

"""

"""

Problem Statements:

Global air travel has seen an upward trend in recent times. The maintenance of operational efficiency and maximizing profitability are crucial for airlines and airport authorities. Businesses need to optimize airline and terminal operations to enhance passenger satisfaction, improve turnover rates, and increase overall revenue.

The airline companies with the available data want to find an opportunity to analyze and understand travel patterns, customer demand, and terminal usage.

CRISP-ML(Q) process model describes six phases:

1. Business and Data Understanding

2. Data Preparation

3. Model Building

4. Model Evaluation

5. Deployment

6. Monitoring and Maintenance

Objective: Maximize the operational efficiency

Constraints: Maximize the financial health

Success Criteria:

Business Success Criteria: Increase the operational efficiency by 10% to 12% by segmenting the Airlines.

ML Success Criteria: Achieve a Silhouette coefficient of at least 0.7

Economic Success Criteria: The airline companies will see an increase in revenues by at least 8% (hypothetical numbers)

"""

pip install py-AutoClean

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import sweetviz

from AutoClean import AutoClean

from sklearn.preprocessing import MinMaxScaler

from sklearn.pipeline import make\_pipeline

from scipy.cluster.hierarchy import linkage, dendrogram

from sklearn.cluster import AgglomerativeClustering

from sqlalchemy import create\_engine, text

traff = pd.read\_csv(r"C:/Users/Lenovo/Downloads/Study material/Data Science/Hierarchical Clustering\_Hands-on/Assignments/Data Set/Data Set (5)/AirTraffic\_Passenger\_Statistics.csv")

user = "root"

pw = 1234

db = "traff\_db"

engine = create\_engine(f"mysql+pymysql://{user}:{pw}@localhost/{db}")

# to\_sql() - function to push the dataframe onto a SQL table.

traff.to\_sql('traff\_tbl', con = engine, if\_exists = 'replace', chunksize = 1000, index = False)

###### To read the data from MySQL Database

sql = 'select \* from traff\_tbl;'

df = pd.read\_sql\_query(text(sql), engine.connect())

# Data types

df.info()

# EXPLORATORY DATA ANALYSIS (EDA) / DESCRIPTIVE STATISTICS

# \*\*\*Descriptive Statistics and Data Distribution Function\*\*\*

df.describe()

# Data Preprocessing

# \*\*Cleaning Unwanted columns\*\*

# UnivID is the identity to each university.

# Analytically it does not have any value (Nominal data).

# We can safely ignore the ID column by dropping the column.

df.drop(['Operating Airline IATA Code', 'Boarding Area', 'Year', 'Month'], axis = 1, inplace = True)

df.info()

# ## Automated Libraries

# AutoEDA

# import sweetviz

my\_report = sweetviz.analyze([df, "df"])

my\_report.show\_html('Report.html')

'''

Alternatively, we can use other AutoEDA functions as well.

# D-Tale

########

pip install dtale

import dtale

d = dtale.show(df)

d.open\_browser()

'''

# EDA report highlights:

# ------------------------

# Missing Data: Identified Missing Data in columns: SAT, GradRate

# Outliers: Detected exceptional values in 4 columns: SAT, Top10, Accept, SFRatio

# Boxplot

#Install PyQt5 if you get this warning message - "UserWarning:Matplotlib is currently using agg, which is a non-GUI backend, so cannot show the figure."

#pip install PyQt5

#import PyQt5

df.plot(kind = 'box', subplots = True, sharey = False, figsize = (15, 8))

# increase spacing between subplots

plt.subplots\_adjust(wspace = 0.75) # ws is the width of the padding between subplots, as a fraction of the average Axes width.

plt.show()

# Encoding: 'State' is categorical data that needs to be encoded into numeric values

# Data Preprocessing

# -----------------------------------------------------------------------------

# Auto Preprocessing and Cleaning

# from AutoClean import AutoClean

clean\_pipeline = AutoClean(df.iloc[:, :], mode = 'manual', missing\_num = 'auto'

#, outliers = 'winz')

, encode\_categ = ['auto'])

# help(AutoClean)

# Missing values = 'auto': AutoClean first attempts to predict the missing values with Linear Regression

# outliers = 'winz': outliers are handled using winsorization

# encode\_categ = 'auto': Label encoding performed (if more than 10 categories are present)

df\_clean = clean\_pipeline.output

df\_clean.head()

df\_clean.info()

from sklearn.preprocessing import LabelEncoder

# Creating instance of labelencoder

labelencoder = LabelEncoder()

# Data Split into Input and Output variables

X = df\_clean.iloc[:, :4]

df\_clean['Activity Period'] = labelencoder.fit\_transform(df\_clean['Activity Period'])

df\_clean['Operating Airline'] = labelencoder.fit\_transform(df\_clean['Operating Airline'])

df\_clean['GEO Region'] = labelencoder.fit\_transform(df\_clean['GEO Region'])

df\_clean['Terminal'] = labelencoder.fit\_transform(df\_clean['Terminal'])

# #### Drawback with this approach: If there are more than 10 categories, then Autoclean performs label encoding.

# df\_clean.drop(['State'], axis = 1, inplace = True)

df\_clean.head()

# -----------------------------------------------------------------------------

# ## Normalization/MinMax Scaler - To address the scale differences

# ### Python Pipelines

# from sklearn.pipeline import make\_pipeline

# from sklearn.preprocessing import MinMaxScaler

df\_clean.info()

cols = list(df\_clean.columns)

print(cols)

pipe1 = make\_pipeline(MinMaxScaler())

cols\_n = list(df\_clean.columns)

# Train the data preprocessing pipeline on data

df\_pipelined = pd.DataFrame(pipe1.fit\_transform(df\_clean), columns = cols\_n, index = df\_clean.index)

df\_pipelined.head()

df\_pipelined.describe() # scale is normalized to min = 0; max = 1

###### End of Data Preprocessing ######

# -----------------------------------------------------------------------------

######### Model Building #########

# # CLUSTERING MODEL BUILDING

# ### Hierarchical Clustering - Agglomerative Clustering

# from scipy.cluster.hierarchy import linkage, dendrogram

# from sklearn.cluster import AgglomerativeClustering

# import matplotlib.pyplot as plt

# get\_ipython().run\_line\_magic('matplotlib', 'inline') --- if running in jupyter notebook

plt.figure(1, figsize = (16, 8))

tree\_plot = dendrogram(linkage(df\_pipelined, method = "complete"))

plt.title('Hierarchical Clustering Dendrogram')

plt.xlabel('Index')

plt.ylabel('Euclidean distance')

plt.show()

# Applying AgglomerativeClustering and grouping data into 3 clusters

# based on the above dendrogram as a reference

hc1 = AgglomerativeClustering(n\_clusters = 14,

#affinity = 'euclidean',

linkage = 'ward')

y\_hc1 = hc1.fit\_predict(df\_pipelined)

y\_hc1

# Analyzing the Results obtained

hc1.labels\_ # Referring to the cluster labels assigned

cluster\_labels = pd.Series(hc1.labels\_)

# Combine the labels obtained with the data

df\_clust = pd.concat([cluster\_labels, df\_clean], axis = 1)

df\_clust.head()

df\_clust.columns

df\_clust = df\_clust.rename(columns = {0: 'cluster'})

df\_clust.head()

# # Clusters Evaluation

# \*\*Silhouette coefficient:\*\*

# Silhouette coefficient is a Metric, which is used for calculating

# goodness of the clustering technique, and the value ranges between (-1 to +1).

# It tells how similar an object is to its own cluster (cohesion) compared to

# other clusters (separation).

# A score of 1 denotes the best meaning that the data point is very compact

# within the cluster to which it belongs and far away from the other clusters.

# Values near 0 denote overlapping clusters.

# from sklearn import metrics

metrics.silhouette\_score(df\_pipelined, cluster\_labels)

'''Alternatively, we can use:'''

# \*\*Calinski Harabasz:\*\*

# Higher value of the CH index means clusters are well separated.

# There is no thumb rule which is an acceptable cut-off value.

metrics.calinski\_harabasz\_score(df\_pipelined, cluster\_labels)

# \*\*Davies-Bouldin Index:\*\*

# Unlike the previous two metrics, this score measures the similarity of clusters.

# The lower the score the better the separation between your clusters.

# Vales can range from zero and infinity

metrics.davies\_bouldin\_score(df\_pipelined, cluster\_labels)

'''Hyperparameter Optimization for Hierarchical Clustering'''

# Experiment to obtain the best clusters by altering the parameters

# ## Cluster Evaluation Library

# pip install clusteval

# Refer to link: https://pypi.org/project/clusteval

# from clusteval import clusteval

# import numpy as np

# Silhouette cluster evaluation.

ce = clusteval(evaluate = 'silhouette')

df\_array = np.array(df\_pipelined)

# Fit

ce.fit(df\_array)

# Plot

ce.plot()

## Using the report from clusteval library building 2 clusters

# Fit using agglomerativeClustering with metrics: euclidean, and linkage: ward

hc\_2clust = AgglomerativeClustering(n\_clusters = 14,

#affinity = 'euclidean',

linkage = 'ward')

y\_hc\_2clust = hc\_2clust.fit\_predict(df\_pipelined)

# Cluster labels

hc\_2clust.labels\_

cluster\_labels2 = pd.Series(hc\_2clust.labels\_)

# Concate the Results with data

df\_2clust = pd.concat([cluster\_labels2, df\_clean], axis = 1)

df\_2clust = df\_2clust.rename(columns = {0:'cluster'})

df\_2clust.head()

pd.set\_option('display.max\_columns', None)

# Aggregate using the mean of each cluster

df\_2clust.iloc[:, 1:].groupby(df\_2clust.cluster).mean()

# Save the Results to a CSV file

df\_3clust = pd.concat([df, cluster\_labels2], axis = 1)

df\_3clust = df\_3clust.rename(columns = {0:'cluster'})

df\_3clust.select\_dtypes(include=['float64', 'int64']).groupby('cluster').mean()

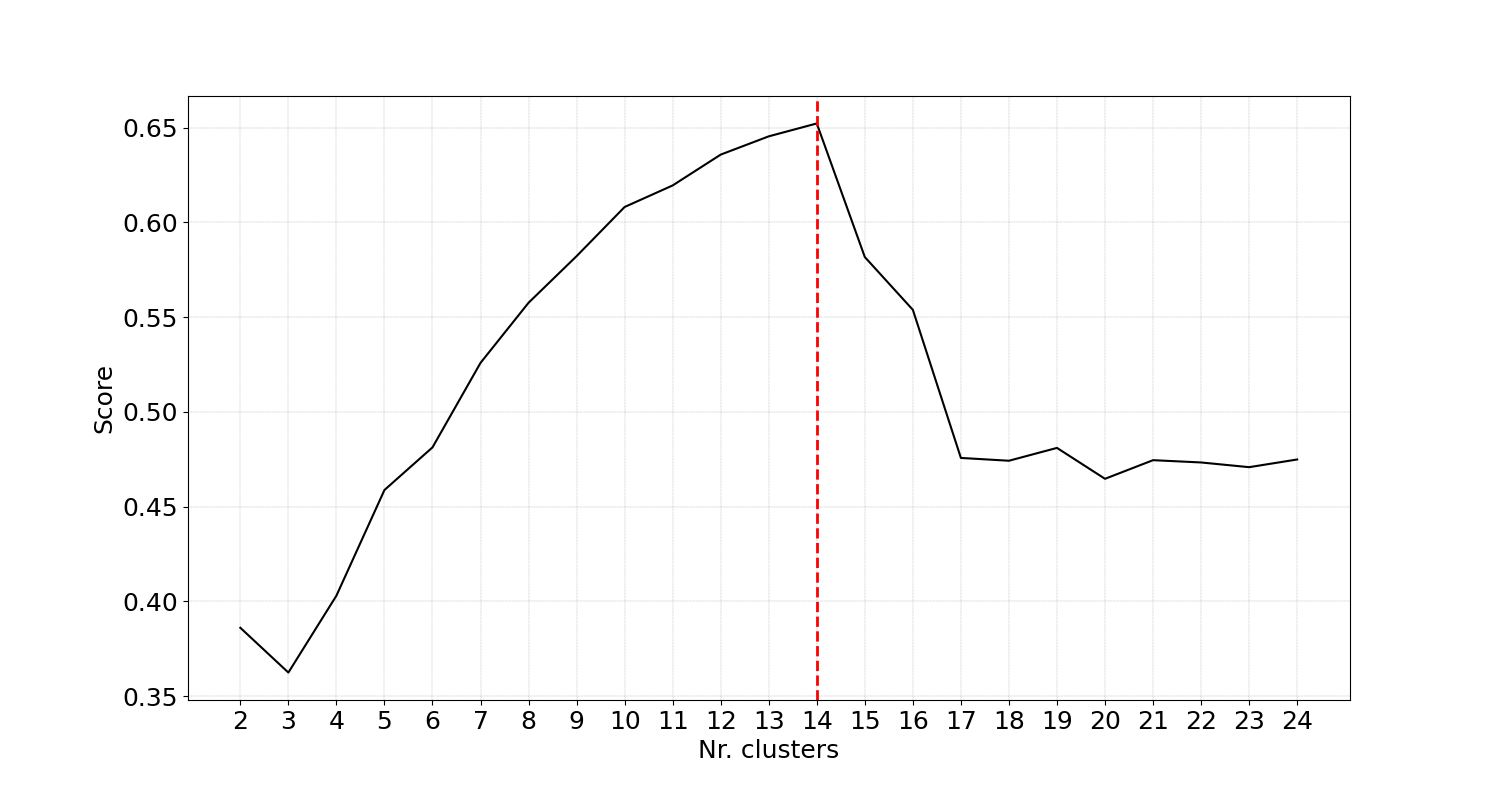
df\_3clust.dtypes

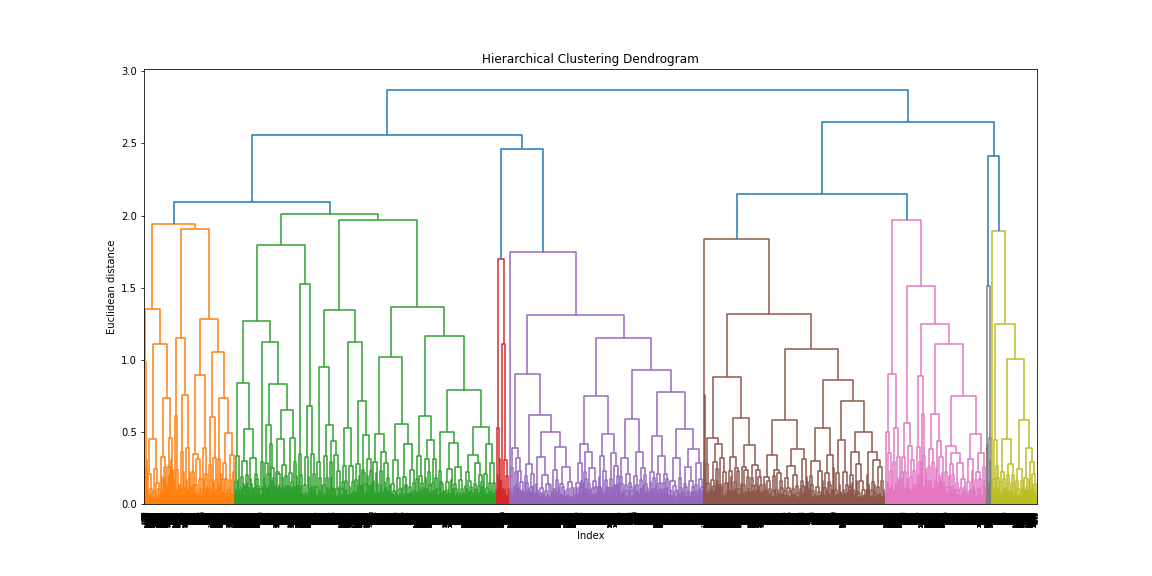
df\_3clust.to\_csv('Airtraffic.csv', encoding = 'utf-8')

import os

os.getcwd()

**Outputs:**





**metrics.silhouette\_score**(df\_pipelined, cluster\_labels)

Out[201]: **0.6521894814361536**

metrics.calinski\_harabasz\_score(df\_pipelined, cluster\_labels)

Out[206]: 9775.728442207464

metrics.davies\_bouldin\_score(df\_pipelined, cluster\_labels)

Out[211]: 0.7932175030681136

**Activity Period Passenger Count**

**cluster**

**0 201046.193706 13435.004583**

**1 201073.937769 12755.652465**

**2 201282.349693 5210.699387**

**3 201028.113445 7509.709034**

**4 200980.453770 35606.647541**

**5 201209.821497 8256.205374**

**6 201121.451951 23812.227402**

**7 200964.814246 119212.472765**

**8 200993.457259 6417.016282**

**9 201241.986842 6920.013158**

**10 201349.681818 128037.174242**

**11 201016.551402 11426.238985**

**12 201072.277372 4946.715328**

**13 200849.736486 6793.398649**

