KARNATAKA LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Course Activity Report on

“ Preprocess a given dataset by applying k-means clustering.”

Submitted in the partial fulfillment for the academic requirement of

5th Semester B.E.

in

Data Warehousing and Data Mining - 21CS543

Submitted by

Shifali Thakur	**2GI21CS142**
Varsha Kannur	**2GI21CS186**
Sanika Karnik	**2GI21CS136**
Rachna Kulkarni	**2GI21CS118**

GUIDE

Prof. Prashant Niranjan

Computer Science Department, Gogte Institute of Technology

2023 – 2024

KARNATAKA LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Ms. Shifali Thakur , Varsha Kannur, Sanika Karnik, Rachana Kulkarni of 5th semester and bearing USN 2GI21CS142, 2GI21CS186 , 2GI21CS136, 2GI21CS118 has satisfactorily completed the course activity (Seminar/Project) in Data Warehousing and Data Mining ***(Course code: 21CS543)*** . It can be considered as a bonafide work carried out in partial fulfillment for the academic requirement of 5th Semester B.E. Computer Science & Engineering prescribed by KLS Gogte Institute of Technology, Belagavi during the academic year **2023- 2024**

The report has been approved as it satisfies the academic requirements in respect of Assignment (Course activity) prescribed for the said Degree.

Signature of the Faculty Member

Signature of the HOD

Marks allocation:

	Batch No. :					
1.	Project Title:	Marks Range	USN			
			2GI21CS142	2GI21CS186	2GI21CS136	2GI21CS118
2.	Problem statement (PO2)	0-1				
3.	Objectives of Defined Problem statement (PO1,PO2)	0-2				
4.	Design / Algorithm/Flowchart/Methodology (PO3)	0-3				
5.	Implementation details/Function/Procedures/Classes and Objects (Language/Tools) (PO1,PO3,PO4,PO5)	0-4				
6.	Working model of the final solution (PO3,PO12)	0-5				
7.	Report and Oral presentation skill (PO9,PO10)	0-5				
	Total	20				
	Reduced to	10				

*** 20 marks is converted to 10 marks for CGPA calculation**

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.

3. Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need
for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological channel

Preprocess a given dataset by applying k-means clustering

Objective

The objective of applying k-means clustering to a dataset during preprocessing is to group similar data points together into clusters. K-means is an unsupervised machine learning algorithm that partitions a dataset into k clusters, where each data point belongs to the cluster with the nearest mean.

1. Choose the Number of Clusters (k): Decide on the number of clusters you want to create in your dataset. This is a critical step, as the choice of k can impact the quality of the clustering results.

2. Feature Selection/Extraction (Optional): Depending on the nature of your dataset, you may choose to select relevant features or perform feature extraction before applying k-means. This step can help improve the quality of clustering.

3. Normalize/Standardize Data (Optional):It's often a good practice to normalize or standardize the data before applying k-means, especially if the features have different scales.

4. Apply K-Means Algorithm: Use the k-means algorithm to cluster the data points into k clusters. The algorithm iteratively assigns data points to clusters based on the mean of the cluster and updates the cluster centroids.

5. Assign Clusters to Data Points: After the k-means algorithm converges, each data point will be

The main objectives of applying k-means clustering during preprocessing are:

- Pattern Recognition: Identify patterns or groups of similarity in the dataset.

- Data Simplification: Reduce the complexity of the dataset by grouping similar data points

Algorithm

Input: Raw dataset (in CSV format)

Output: Model evaluation metrics (accuracy, confusion matrix, classification report)

1. Load and Inspect Data:

   a. Load the dataset using pandas.

   b. Display the first few rows of the dataset.

2. Data Preprocessing:

   a. Handle missing values by filling with the mean of the respective column.

   b. Encode categorical variables using LabelEncoder.

   c. Select relevant features for clustering (feature_columns).

   d. Apply k-means clustering with a specified number of clusters (n_clusters).

3. Visualize Clusters:

   a. Use PCA for dimensionality reduction.

   b. Plot the clustered data points using matplotlib.

   c. Display centroids on the plot.

4. Prepare Data for Machine Learning:

   a. Split the data into features (X) and the target variable (y).

   b. Split the dataset into training and testing sets using train_test_split.

5. Standardize Features:Optionally, standardize the features using StandardScaler.

6. Machine Learning Model (Logistic Regression):

   a. Instantiate a Logistic Regression model.

   b. Train the model on the training set (X_train, y_train).

   c. Make predictions on the test set (X_test).

7. Evaluate the Model: Calculate accuracy, confusion matrix, and classification report for model evaluation. Display the evaluation metrics.

Methodology:

1. Load the Dataset: Use a library like pandas to load the dataset into a DataFrame.

2. Handle Missing Values: Identify columns with missing values. Decide on a strategy to handle missing values (e.g., fill with mean, median, or use more advanced imputation techniques). Implement the chosen strategy to fill or drop missing values.

3. Encode Categorical Variables :Identify categorical columns in the dataset. Choose an encoding method (e.g., Label Encoding or One-Hot Encoding). Apply the selected encoding method to convert categorical variables into numerical representations.

4. Select Relevant Features for Clustering: Decide on the features that are relevant for k-means clustering. Create a new Data Frame containing only the selected features.

5. Apply K-Means Clustering: Choose the number of clusters (k) based on the characteristics of the data. Apply the k-means clustering algorithm to the selected features. Assign cluster labels to each data point.

6. Visualize Clusters Using PCA: Use PCA (Principal Component Analysis) for dimensionality reduction. Fit PCA to the clustered data. Plot the data points in a 2D or 3D space using the first few principal components. Visualize clusters using different colours for each cluster. Optionally, plot centroids of the clusters.

7. Generate Pre-processed Features: Add the cluster labels as a new feature to the original dataset.If needed, create additional features based on the clustering results.

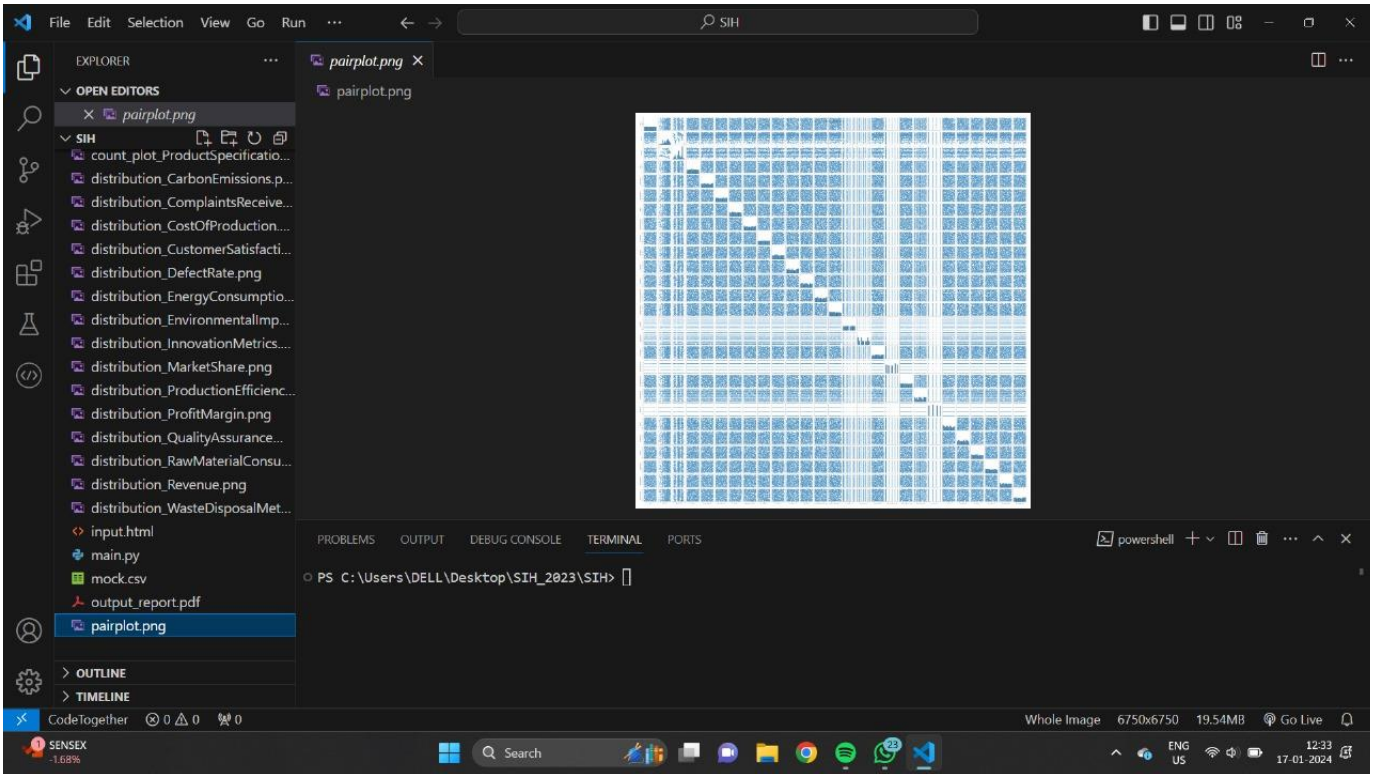
8. Further Analysis: Depending on the goals of your analysis, explore the impact of the generated features on other tasks. Consider using the pre-processed dataset for machine learning tasks, such as classification, regression, or anomaly detection.

Implementation Code:

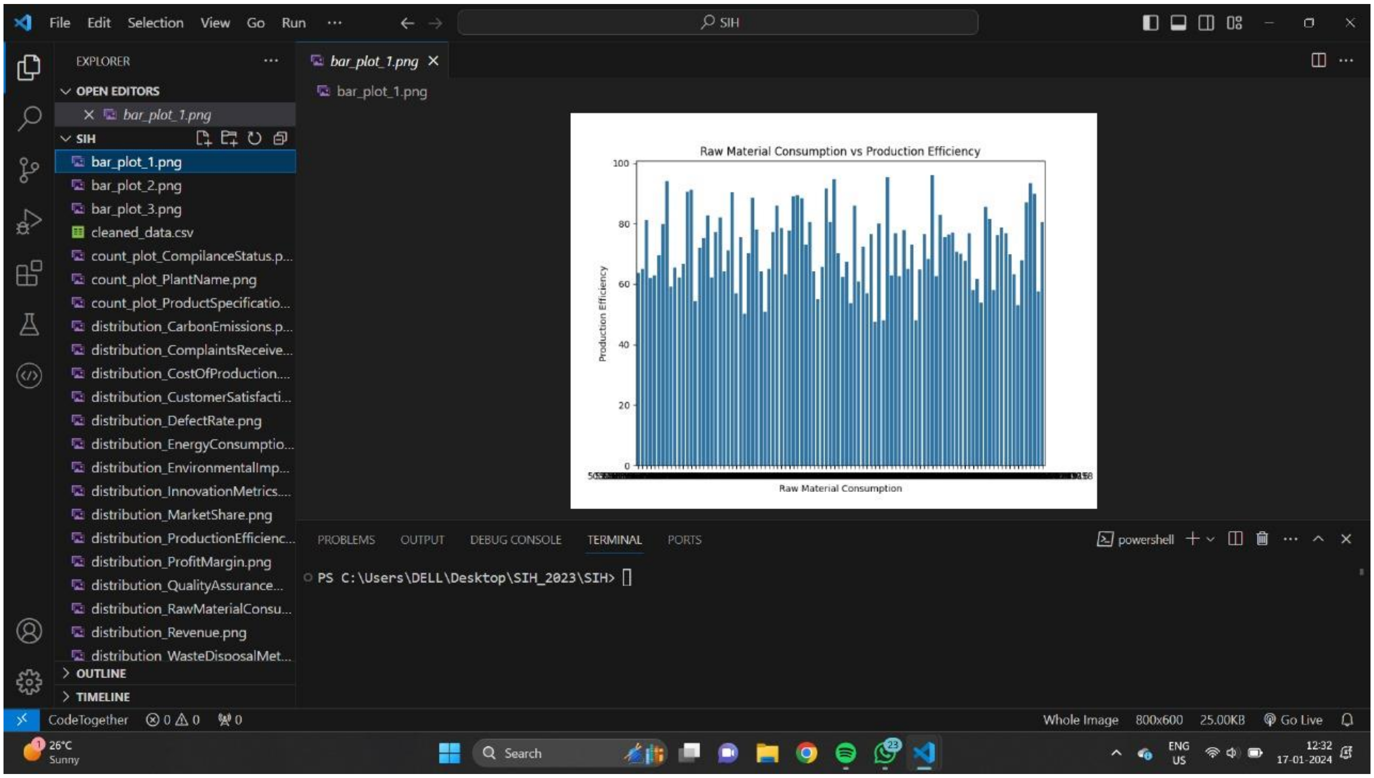
```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Load your dataset (replace 'your_dataset.csv' with the actual file path or URL)
data = pd.read_csv('your_dataset.csv')

# Display the first few rows of the dataset
print("Original Dataset:")
print(data.head())

# Handle missing values (replace 'column_name' with the actual column name)
data['column_name'].fillna(data['column_name'].mean(), inplace=True)

# Encode categorical variables (replace 'categorical_column' with the actual column name)
le = LabelEncoder()
data['categorical_column'] = le.fit_transform(data['categorical_column'])

# Combine features for clustering (replace 'feature_columns' with the actual feature columns)
cluster_data = data[['feature_column1', 'feature_column2']]

# Apply k-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
data['cluster'] = kmeans.fit_predict(cluster_data)

# Visualize clusters using PCA for dimensionality reduction
pca = PCA(n_components=2)
cluster_data_pca = pca.fit_transform(cluster_data)

# Plot the clustered data
plt.scatter(cluster_data_pca[:, 0], cluster_data_pca[:, 1], c=data['cluster'], cmap='viridis', edgecolor='k', s=50)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', marker='X', s=200, label='Centroids')
plt.title('K-Means Clustering on Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.show()
```

```
# Split the data into features and target variable
X = data.drop(['target_variable', 'cluster'], axis=1)
y = data['target_variable']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Standardize the features (optional but often beneficial)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Display the preprocessed data
print("\nPreprocessed Dataset:")
print(X_train[:5])  # Displaying the first 5 rows of the preprocessed features
print(y_train[:5])  # Displaying the first 5 rows of the target variable
```

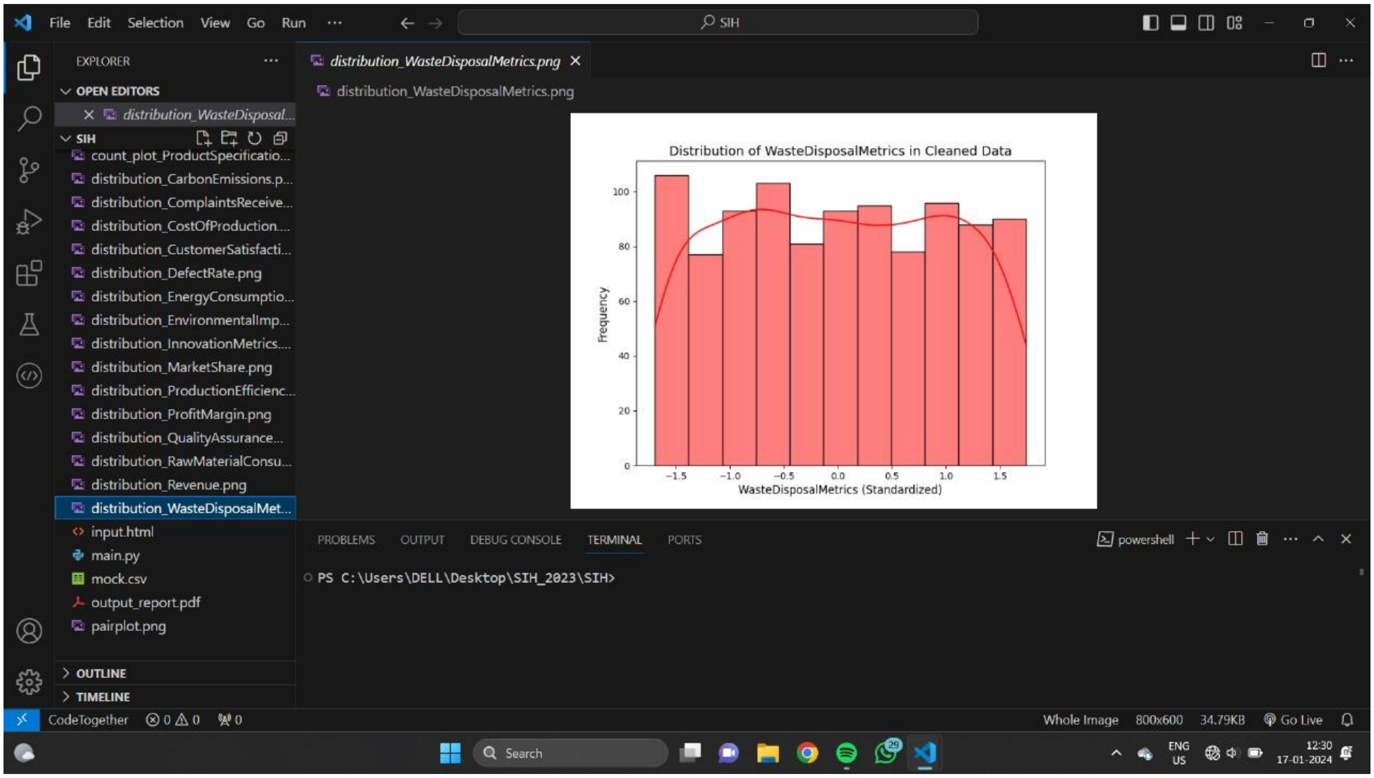
Output:


File
Edit
Selection
View
Go
Run
SIH
EXPLORER
OPEN EDITORS
pairplot.png
SIH
count_plot_ProductSpecificatio...
distribution_CarbonEmissions.p...
distribution_ComplaintsReceive...
distribution_CostOfProduction....
distribution_CustomerSatisfacti...
distribution_DefectRate.png
distribution_EnergyConsumptio...
distribution_EnvironmentalImp...
distribution_InnovationMetrics....
distribution_MarketShare.png
distribution_ProductionEfficienc...
distribution_ProfitMargin.png
distribution_QualityAssurance...
distribution_RawMaterialConsu...
distribution_Revenue.png
distribution_WasteDisposalMet...
input.html
main.py
mock.csv
output_report.pdf
pairplot.png
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
CodeTogether
Whole Image
6750x6750
19.54MB
Go Live
SENSEX
-1.68%
Search
ENG
US
12:33
17-01-2024

```
PS C:\Users\DELL\Desktop\SIH_2023\SIH> 
```


File
Edit
Selection
View
Go
Run
SIH
EXPLORER
OPEN EDITORS
bar_plot_1.png
SIH
bar_plot_1.png
bar_plot_2.png
bar_plot_3.png
cleaned_data.csv
count_plot_CompilanceStatus.p...
count_plot_PlantName.png
count_plot_ProductSpecificatio...
distribution_CarbonEmissions.p...
distribution_ComplaintsReceive...
distribution_CostOfProduction....
distribution_CustomerSatisfacti...
distribution_DefectRate.png
distribution_EnergyConsumptio...
distribution_EnvironmentalImp...
distribution_InnovationMetrics....
distribution_MarketShare.png
distribution_ProductionEfficienc...
distribution_ProfitMargin.png
distribution_QualityAssurance...
distribution_RawMaterialConsu...
distribution_Revenue.png
distribution_WasteDisposalMet...
OUTLINE
TIMELINE
Raw Material Consumption vs Production Efficiency
Production Efficiency
0
20
40
60
80
100
Raw Material Consumption
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
CodeTogether
Whole Image
800x600
25.00KB
Go Live
26°C
Sunny
Search
ENG
US
12:32
17-01-2024

```
PS C:\Users\DELL\Desktop\SIH_2023\SIH> 
```



File
Edit
Selection
View
Go
Run
SIH
EXPLORER
OPEN EDITORS
distribution_WasteDisposal...
SIH
count_plot_ProductSpecificatio...
distribution_CarbonEmissions.p...
distribution_ComplaintsReceive...
distribution_CostOfProduction....
distribution_CustomerSatisfacti...
distribution_DefectRate.png
distribution_EnergyConsumptio...
distribution_EnvironmentalImp...
distribution_InnovationMetrics....
distribution_MarketShare.png
distribution_ProductionEfficienc...
distribution_ProfitMargin.png
distribution_QualityAssurance...
distribution_RawMaterialConsu...
distribution_Revenue.png
distribution_WasteDisposalMet...
input.html
main.py
mock.csv
output_report.pdf
pairplot.png
OUTLINE
TIMELINE
distribution_WasteDisposalMetrics.png
distribution_WasteDisposalMetrics.png
Distribution of WasteDisposalMetrics in Cleaned Data
Frequency
0
20
40
60
80
100
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
WasteDisposalMetrics (Standardized)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
CodeTogether
⊗ 0 ⚠ 0
0
Whole Image
800x600
34.79KB
Go Live
Search
ENG
US
12:30
17-01-2024

```
PS C:\Users\DELL\Desktop\SIH_2023\SIH>
```

Conclusion:

In summary, the application of k-means clustering to preprocess a given dataset facilitates the identification of natural patterns and the segmentation of data into distinct groups. This clustering approach enhances our understanding of the inherent structure within the dataset, offering valuable insights for decision-making and subsequent analyses. By assigning cluster labels to data points, the technique not only aids in feature engineering but also opens avenues for further exploration and optimization. The clustering results contribute to a more organized and nuanced perspective on the dataset, laying the foundation for downstream tasks and providing an opportunity for meaningful analysis and interpretation. Regular documentation of the preprocessing steps and transparent reporting ensure reproducibility and clarity in communicating the outcomes of the clustering process.