Clustering Analysis of Demographic Characteristics of Tamil Nadu Marginal Workers

Introduction:

Demographic characteristics of marginal workers in Tamil Nadu provide important insights into the composition of this labour force and are significant for various reasons, including economic and social policy planning.

• AGE: The age distribution of marginal workers can indicate the youthfulness or aging of this labour force. A higher proportion of young marginal workers may reflect the need for skill development and employment opportunities for the youth.

• SEX: Understanding the gender balance among marginal workers is essential for addressing gender-related labour market disparities.

• INDUSTRIAL CATEGORY: The industrial category or the sector in which the marginal workers work provides an insight about their economy

Objective:

To identify patterns among different industrial categories and age groups among marginal workers based on demographics.

Data Collection and Preparation

Data Source:

https://tn.data.gov.in/resource/marginal-workers-classified-age-industrial-category-and-sex-scheduled-caste-2011-tamil

Data Description:

The attributes included in the dataset are age, gender, and industrial category.

Choosing the Number of Clusters (K):

Using the Elbow method we assign number of clusters, choose an appropriate K value based on the Elbow Method

Hierarchal Clustering Algorithm:

Hierarchical clustering is a type of clustering algorithm used in data analysis and machine learning to create a hierarchy of clusters. It organizes marginal workers data into a hierarchy of clusters, forming a tree-like structure called a dendrogram.

Initialisation:

Initialize N cluster. Each data point starts as a single cluster.

We use complete linkage method to compute the distance between clusters as the maximum distance between their data points. Euclidean distance is used as the distance metric in this clustering process.

Implementation

Software and Libraries:

The software and libraries used for the analysis:

Python

Pandas

Numpy

Matplotlib

Scipy

Seaborn

Code:

import pandas as pd

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

selected\_columns = data.columns[7:-1]

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for K-Means)

scaler = StandardScaler()

normalized\_data = scaler.fit\_transform(selected\_data)

# Determine the optimal number of clusters (K) using the Elbow method

inertia = []

k\_values = range(1, 6)  # Reducing the range to avoid the error

for k in k\_values:

    kmeans = KMeans(n\_clusters=k, random\_state=42)

    kmeans.fit(normalized\_data)

    inertia.append(kmeans.inertia\_)

# Plot the Elbow method graph

plt.figure(figsize=(8, 4))

plt.plot(k\_values, inertia, marker='o', linestyle='--')

plt.title('Elbow Method for Optimal K')

plt.xlabel('Number of Clusters (K)')

plt.ylabel('Inertia (Within-cluster Sum of Squares)')

plt.xticks(k\_values)

plt.grid()

plt.tight\_layout()

plt.show()

# Based on the Elbow method, choose an appropriate K value (number of clusters)

# You can visually inspect the graph and select a value where the inertia starts to level off.

# Perform K-Means clustering with the chosen K value

k = 3  # You can adjust this value based on the Elbow method result

kmeans = KMeans(n\_clusters=k, random\_state=42)

clusters = kmeans.fit\_predict(normalized\_data)

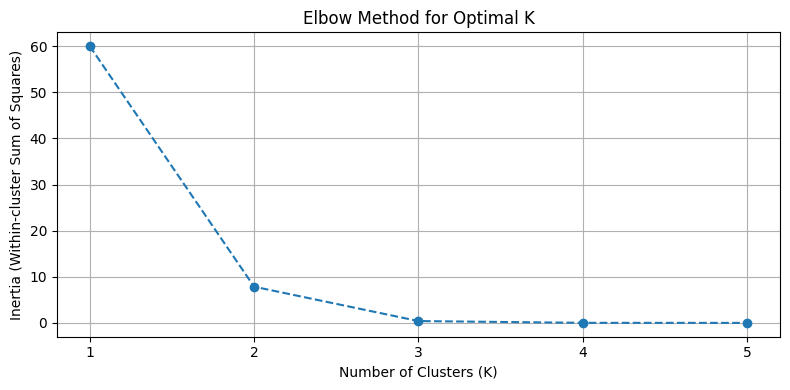
# Add the cluster labels to the DataFrame

data['Cluster'] = clusters

# Now, you have assigned clusters to each row in your dataset.

# You can further analyze and visualize the clustering results as needed.

# For example, you can create scatter plots or other visualizations to see how data points in different clusters are distributed.



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.cluster import hierarchy

from scipy.spatial.distance import pdist

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

# In this case, you may want to use the columns representing industrial categories and age groups.

selected\_columns = data.columns[7:-1]  # Adjust the column range as needed

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for hierarchical clustering)

normalized\_data = (selected\_data - selected\_data.mean()) / selected\_data.std()

# Calculate the pairwise distances between data points

distances = pdist(normalized\_data, metric='euclidean')

# Perform hierarchical clustering

linkage\_matrix = hierarchy.linkage(distances, method='ward')

# Create a cluster diagram (dendrogram)

plt.figure(figsize=(10, 6))

dendrogram = hierarchy.dendrogram(linkage\_matrix, labels=data['Age group'].tolist(), orientation='top')

plt.title('Hierarchical Clustering Dendrogram')

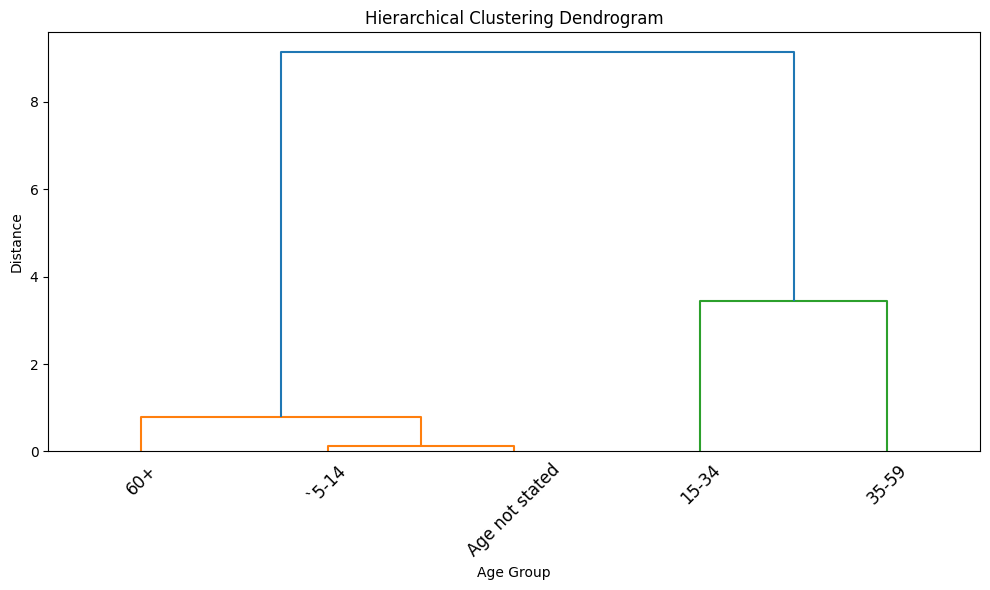
plt.xlabel('Age Group')

plt.ylabel('Distance')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.cluster import hierarchy

from scipy.spatial.distance import pdist

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

# In this case, you may want to use the columns representing industrial categories and age groups.

selected\_columns = data.columns[7:-1]  # Adjust the column range as needed

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for hierarchical clustering)

normalized\_data = (selected\_data - selected\_data.mean()) / selected\_data.std()

# Calculate the pairwise distances between data points

distances = pdist(normalized\_data, metric='euclidean')

# Perform Agglomerative hierarchical clustering

agglomerative\_linkage = hierarchy.linkage(distances, method='ward')

# Create an Agglomerative cluster diagram (dendrogram)

plt.figure(figsize=(10, 6))

dendrogram\_agg = hierarchy.dendrogram(agglomerative\_linkage, labels=data['Age group'].tolist(), orientation='top')

plt.title('Agglomerative Hierarchical Clustering Dendrogram --bottom-up approach--')

plt.xlabel('Age Group')

plt.ylabel('Distance')

plt.xticks(rotation=45)

plt.tight\_layout()

# Perform Divisive hierarchical clustering (top-down approach)

# You can use the fcluster function with t parameter to set the desired number of clusters.

from scipy.cluster.hierarchy import fcluster

k = 3  # Specify the number of clusters for Divisive clustering

divisive\_clusters = fcluster(agglomerative\_linkage, k, criterion='maxclust')

# Add the divisive cluster labels to the DataFrame

data['Divisive Cluster'] = divisive\_clusters

# Now, you can analyze and visualize the divisive clusters as needed.

# Show the Divisive clustering results

print("Divisive Clusters:")

print(data[['Age group', 'Divisive Cluster']])

# Plot the Divisive clusters

plt.figure(figsize=(8, 4))

plt.scatter(data['Divisive Cluster'], data['Age group'], c=data['Divisive Cluster'], cmap='rainbow')

plt.title('Divisive Clustering---top-down approach---')

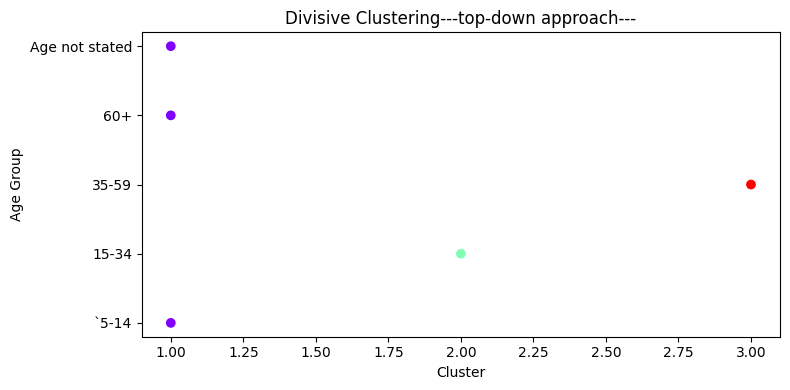
plt.xlabel('Cluster')

plt.ylabel('Age Group')

plt.tight\_layout()

plt.show()





Conclusion:

In conclusion, the clustering analysis of demographic characteristics of Tamil Nadu's marginal workers using age and industrial category has provided valuable insights. This analysis has revealed distinct groups within this workforce based on these two key variables. These clusters can help policymakers and researchers better understand the diverse needs and challenges faced by different segments of marginal workers in the state.

This analysis serves as a foundation for evidence-based decision-making, with the goal to improve the socio-economic activities of Tamil Nadu's marginal workers.