TN Marginal Workers Assessment

Team Member

Name : **ASWIN S**

Register Number : **211521243026**

Applied Data Science Phase-2 document

Team Members :

1. **ASWIN S**
2. **A R HRUDAYABHIRAM**
3. **DINESH S**
4. **LAKSHMI KANTH**
5. **HARIHARAN R**

Phase 2: Innovation

Dataset Link: https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states

Problem Statement:

In this section you need to put your design into innovation to solve the problem. Create a document around it and share the same for assessment as per the instructions mentioned.

CONSIDER CONDUCTING CLUSTERING ANALYSIS TO IDENTIFY PATTERNS AMONG DIFFERENT INDUSTRIAL CATEGORIES AND AGE GROUPS.

Clustering Analysis for Industrial Categories and Age Groups

Introduction

The objective of this analysis is to identify patterns and clusters within different industrial categories and age groups. By conducting a clustering analysis, we aim to gain insights that can inform targeted policies and interventions.

Objective

The main objective of this analysis is to uncover underlying patterns among industrial categories and age groups to facilitate data-driven decision-making for policy formulation and resource allocation.

Methodology

Data Collection

The dataset used for this analysis consists of information pertaining to various industrial categories, age groups, and corresponding workforce statistics. The dataset includes fields such as 'State Code', 'District Code', 'Area Name', 'Age Group', and various industrial category-related metrics.

Data Preprocessing

- Data cleaning: Checked for missing values and inconsistencies, and applied necessary imputation or removal.

- Feature selection: Extracted relevant columns for clustering analysis, including industrial category-related metrics and age group.
- Standardization: Scaled the data to ensure all features have the same scale.

Clustering Algorithm Selection

The **K-Means clustering algorithm** was chosen for this analysis due to its simplicity, efficiency, and suitability for this type of dataset. K-Means is well-suited for identifying clusters in numerical data.

Determining Number of Clusters

The **Elbow Method** was employed to determine the optimal number of clusters. By plotting the sum of squared distances for different numbers of clusters, we identified the 'elbow point' which indicated an optimal number of clusters.

Clustering Analysis

The K-Means algorithm was applied to the standardized data with the determined number of clusters. Each data point was assigned to a specific cluster based on its characteristics.

Step 1: Data Collection and Preparation

1. **Data Collection**:
    - Obtain a dataset containing information on industrial categories, age groups, and relevant workforce statistics. This dataset should ideally include fields such as 'State Code', 'District Code', 'Area Name', 'Age Group', and various industrial category-related metrics.

2. **Data Cleaning and Exploration**:
    - Start by inspecting the dataset for any missing values, outliers, or inconsistencies. Address these issues through imputation, removal, or other appropriate methods. Explore the data to gain a preliminary understanding of its structure and variables.
3. **Feature Selection**:
    - Identify the columns related to industrial categories and age groups. These will be the primary features used in the clustering analysis.
4. **Standardization**:
    - Since you're dealing with different types of metrics (e.g., workforce counts, percentages), consider standardizing the data to ensure all features are on a similar scale. This is important for the effectiveness of the clustering algorithm.

Step 2: Choosing the Clustering Algorithm

1. **Selecting a Clustering Algorithm**:
    - Given that you're dealing with numerical data, K-Means clustering is a suitable choice. It's efficient and effective for identifying clusters in numeric data.

Step 3: Determining the Number of Clusters

1. **Elbow Method**:
    - Use the Elbow Method to find the optimal number of clusters. This involves running the K-Means algorithm for a range of cluster numbers and plotting the sum of squared distances. The "elbow

point" in the plot represents an optimal balance between accuracy and simplicity.

Step 4: Applying Clustering Algorithm

1. **Standardizing Data**:
   - If not done earlier, standardize the relevant columns related to industrial categories and age groups.
2. **Applying K-Means Algorithm**:
   - Apply the K-Means algorithm with the determined number of clusters. This will assign each data point to a specific cluster based on the features related to industrial categories and age groups.

Step 5: Visualizing Clusters

1. **Scatter Plots**:
   - Create scatter plots to visualize the clusters. Each data point will be represented on the plot, with colors indicating the assigned cluster. Since you're working with two dimensions (age groups and industrial categories), you can create scatter plots that show the relationship between these variables.

Step 6: Interpreting Clusters

1. **Analyzing Clusters**:
   - Examine the clusters to understand the common characteristics that group data points together. In

this case, look at how age groups and industrial categories are grouped within each cluster.

2. **Extracting Insights**:
   - Derive meaningful insights from the identified patterns. Consider how these patterns can inform policies or interventions related to workforce dynamics in different industrial categories and age groups.

Step 7: Recommendations and Conclusion

1. **Recommendations**:
   - Based on the clustering results, provide actionable recommendations for policies or strategies related to workforce dynamics in specific industrial categories and age groups. Tailor these recommendations to address the unique needs of each cluster.
2. **Conclusion**:
   - Summarize the key findings and emphasize the value of the clustering analysis in providing data-driven insights for policy-making.

Code

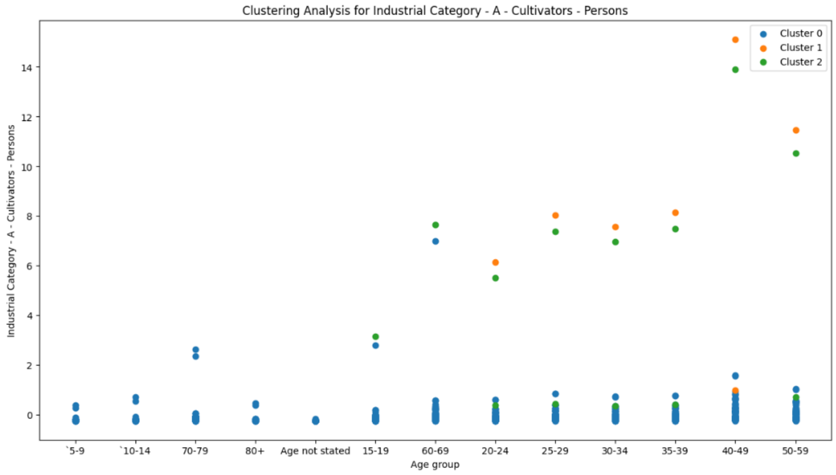
GOOGLE COLLAB LINK:

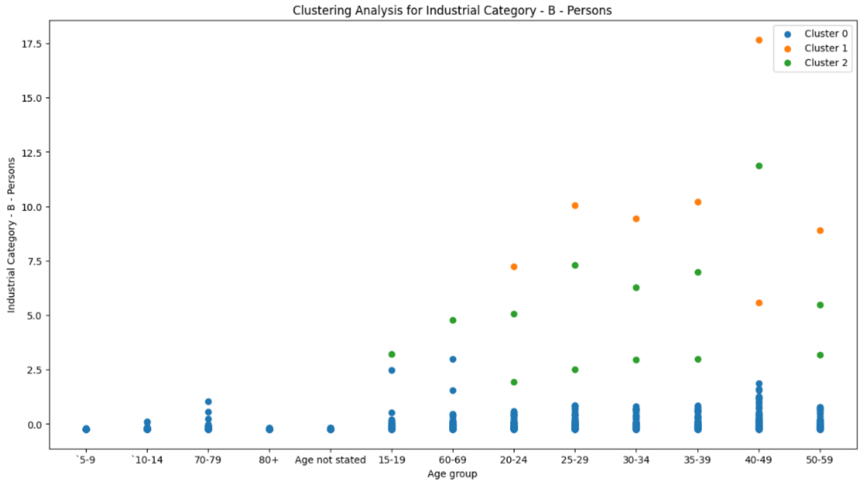
https://colab.research.google.com/drive/1aNDzzS4MtE4NzUstX7GNCLs8OavTQikK?usp=sharing

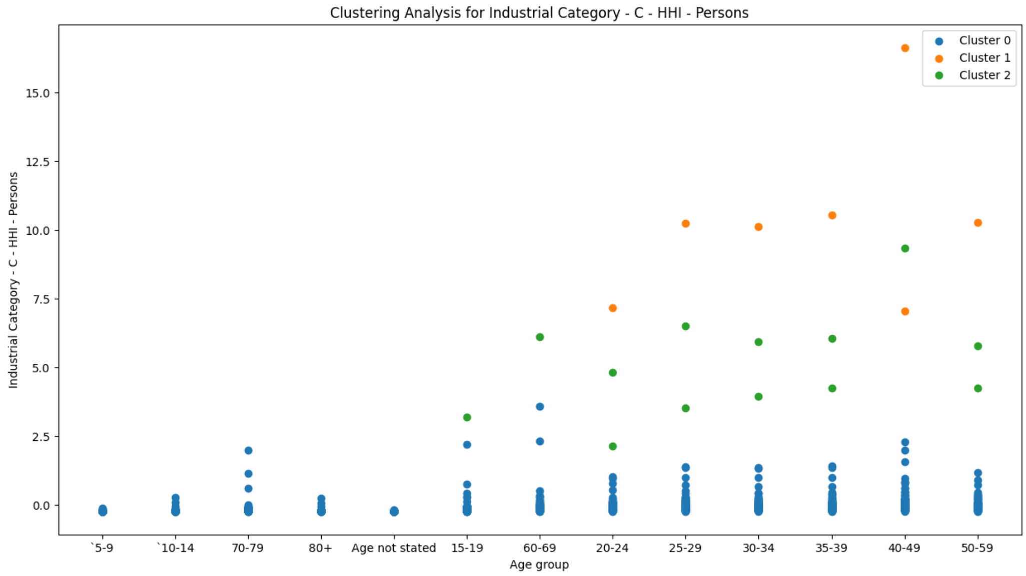
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

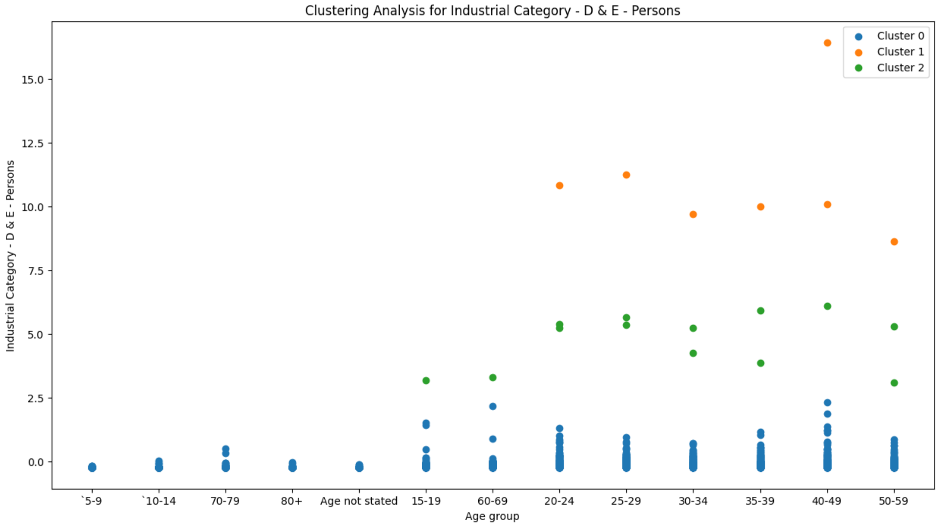
```
# Load your dataset into a DataFrame (replace 'your_dataset_path' with the actual path)
df = pd.read_csv('marginal_workers_tamil_nadu.csv.csv')

# Data Cleaning and Exploration
numeric_columns = ['Industrial Category - A - Cultivators - Persons',
                   'Industrial Category - B - Persons',
                   'Industrial Category - C - HHI - Persons',
                   'Industrial Category - D & E - Persons',
                   'Industrial Category - F - Persons',
                   'Industrial Category - G - HHI - Persons',
                   'Industrial Category - G - Non HHI - Persons',
                   'Industrial Category - H - Persons',
                   'Industrial Category - I - Persons',
                   'Industrial Category - J - HHI - Persons',
                   'Industrial Category - J - Non HHI - Persons',
                   'Industrial Category - K to M - Persons',
                   'Industrial Category - N to O - Persons',
                   'Industrial Category - P to Q - Persons',
                   'Industrial Category - R to U - HHI - Persons',
                   'Industrial Category - R to U - Non HHI - Persons'
                   ]

# Convert columns to numeric, handling errors as NaN
df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric, errors='coerce')

# Drop rows with NaN values in numeric columns
df.dropna(subset=numeric_columns, inplace=True)

# Exclude rows where 'Age group' is "Total"
df = df[df['Age group'] != 'Total']

# Standardizing Data
scaler = StandardScaler()
features = numeric_columns[:]  # Use all the industrial categories
df[features] = scaler.fit_transform(df[features])

# Applying K-Means Algorithm
n_clusters = 3  # Assuming you've determined the optimal number of clusters
kmeans = KMeans(n_clusters=n_clusters, random_state=0)
df['Cluster'] = kmeans.fit_predict(df[features])

# Scatter Plots for All Industrial Categories
for feature in features:
    plt.figure(figsize=(15, 8))
    for cluster in range(n_clusters):
        cluster_df = df[df['Cluster'] == cluster]
        plt.scatter(cluster_df['Age group'], cluster_df[feature],
label=f'Cluster {cluster}')

    plt.xlabel('Age group')
    plt.ylabel(feature)
    plt.title(f'Clustering Analysis for {feature}')
    plt.legend()
    plt.show()

    plt.savefig(f'cluster_plot_{feature}.png')
    plt.close()


# Further steps for analysis and recommendations can be done here
```

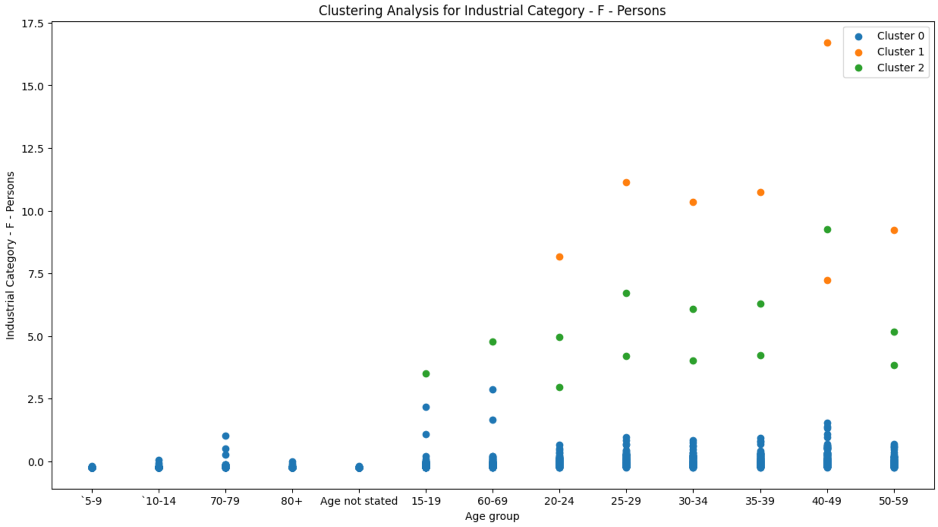
OUTPUT:

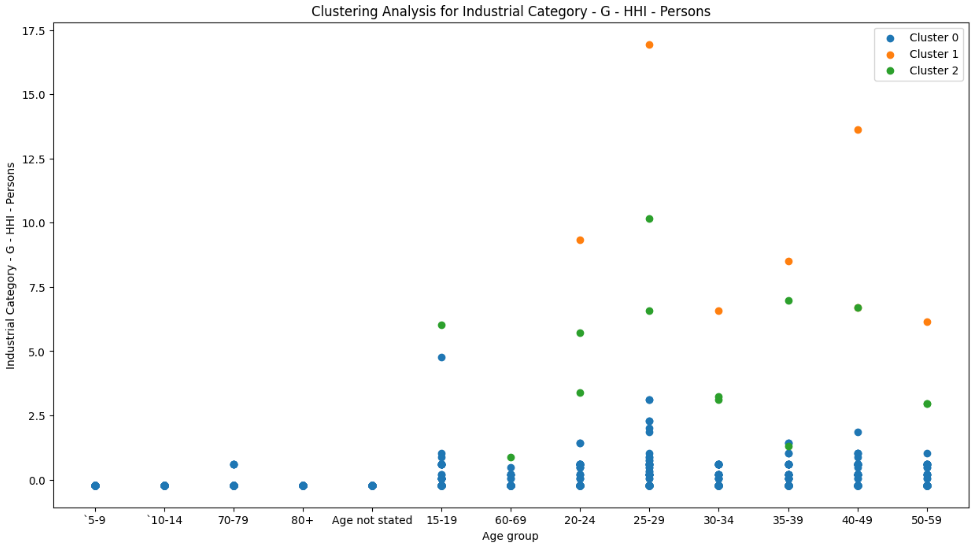

Clustering Analysis for Industrial Category - A - Cultivators - Persons
Cluster 0
Cluster 1
Cluster 2
Industrial Category - A - Cultivators - Persons
0
2
4
6
8
10
12
14
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group

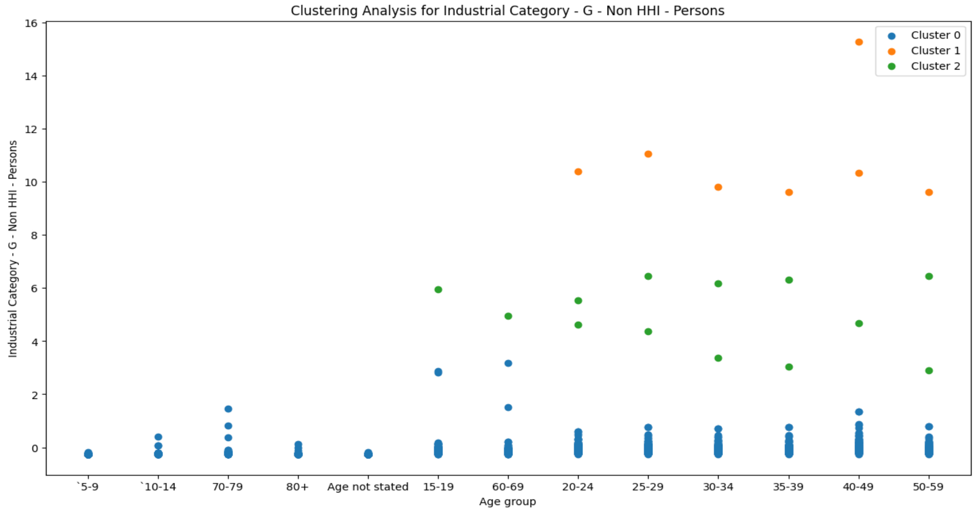

Clustering Analysis for Industrial Category - B - Persons
Cluster 0
Cluster 1
Cluster 2
Industrial Category - B - Persons
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group

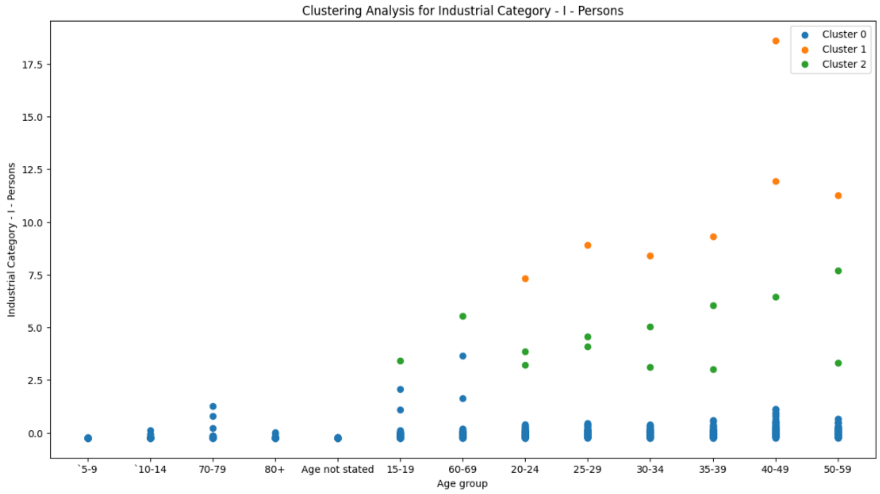

Clustering Analysis for Industrial Category - C - HHI - Persons
Cluster 0
Cluster 1
Cluster 2
Industrial Category - C - HHI - Persons
0.0
2.5
5.0
7.5
10.0
12.5
15.0
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group

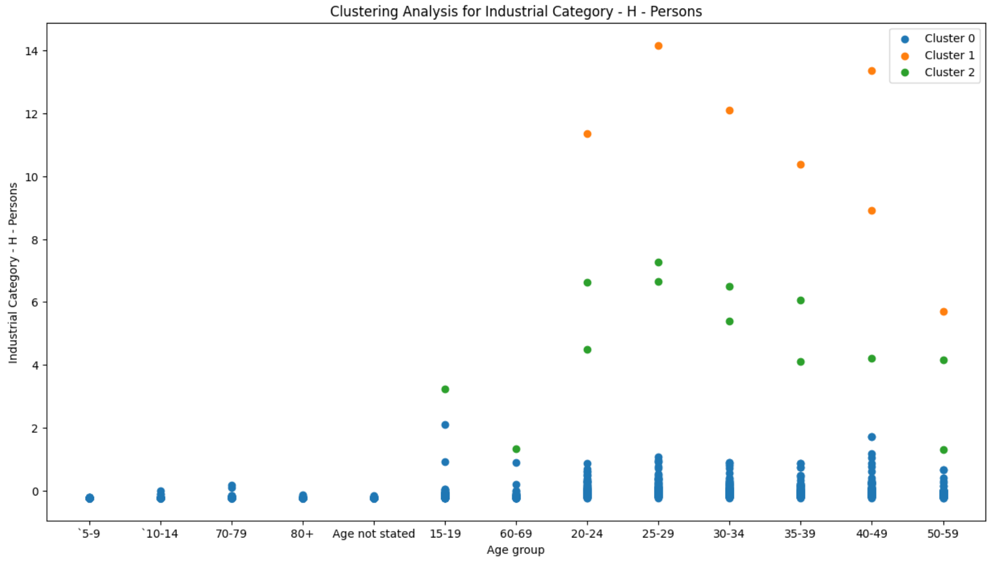

Clustering Analysis for Industrial Category - D & E - Persons
Industrial Category - D & E - Persons
15.0
12.5
10.0
7.5
5.0
2.5
0.0
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group
Cluster 0
Cluster 1
Cluster 2


Clustering Analysis for Industrial Category - F - Persons
Industrial Category - F - Persons
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group
Cluster 0
Cluster 1
Cluster 2

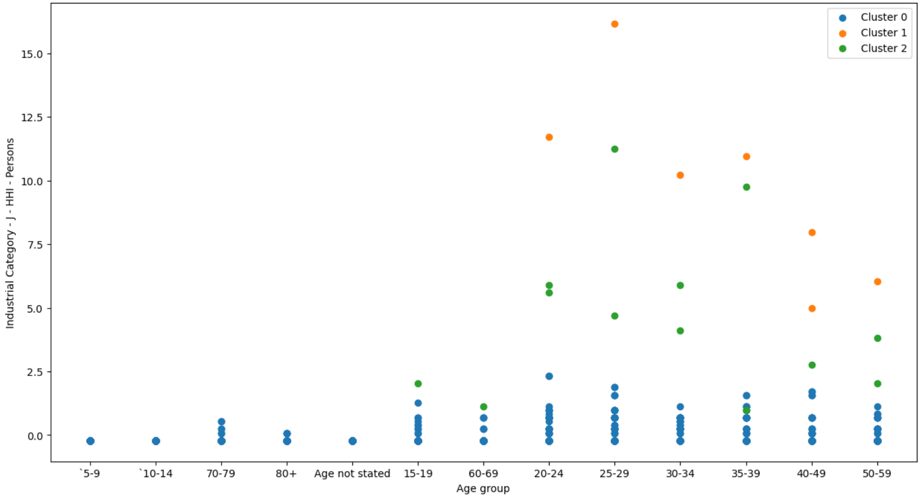

Clustering Analysis for Industrial Category - G - HHI - Persons
Industrial Category - G - HHI - Persons
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
Age group
Cluster 0
Cluster 1
Cluster 2


Clustering Analysis for Industrial Category - G - Non HHI - Persons
Industrial Category - G - Non HHI - Persons
Age group
0
2
4
6
8
10
12
14
16
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
● Cluster 0
● Cluster 1
● Cluster 2

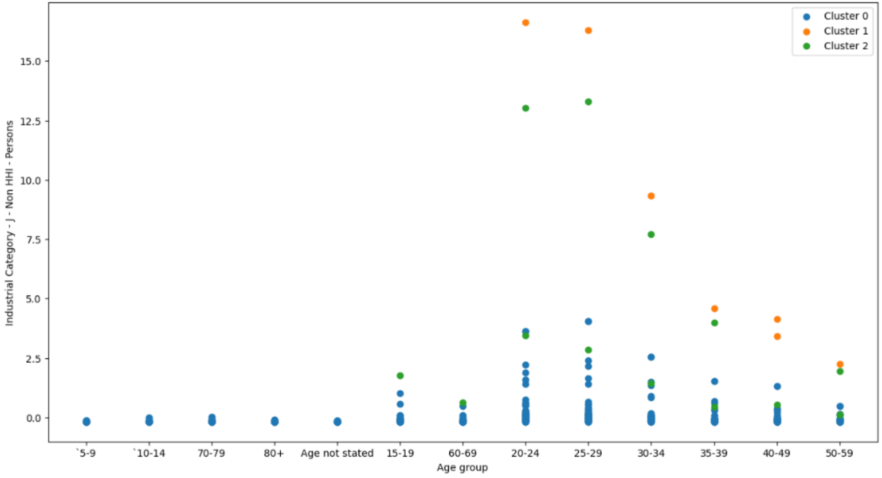

Clustering Analysis for Industrial Category - I - Persons
Industrial Category - I - Persons
Age group
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
● Cluster 0
● Cluster 1
● Cluster 2


Clustering Analysis for Industrial Category - H - Persons
Industrial Category - H - Persons
Age group
0
2
4
6
8
10
12
14
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59
● Cluster 0
● Cluster 1
● Cluster 2

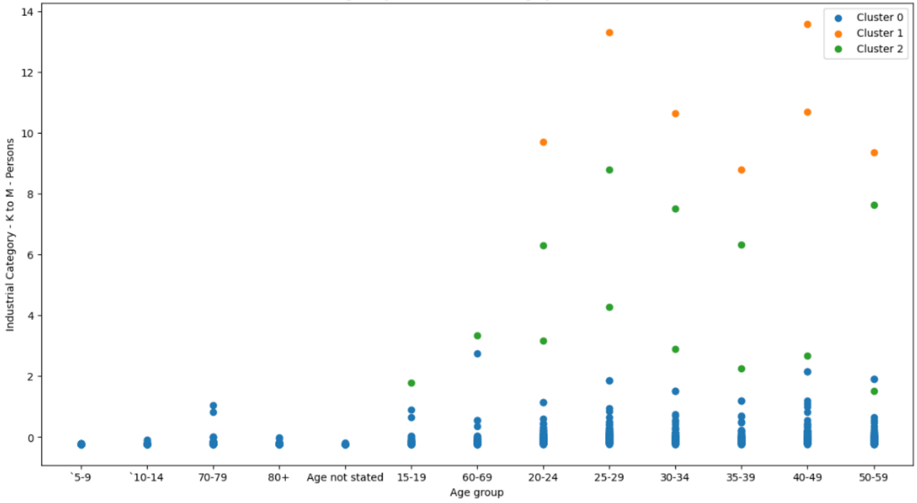
Clustering Analysis for Industrial Category - J - HHI - Persons

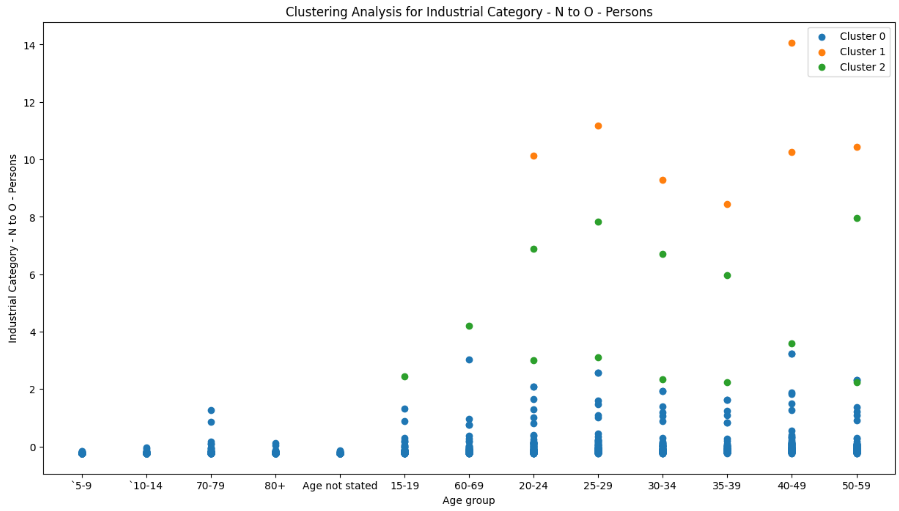


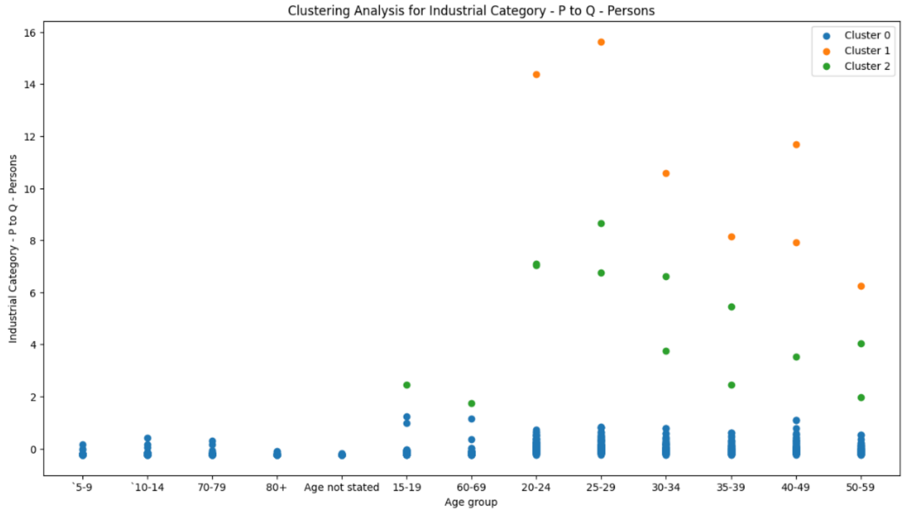
Clustering Analysis for Industrial Category - J - Non HHI - Persons

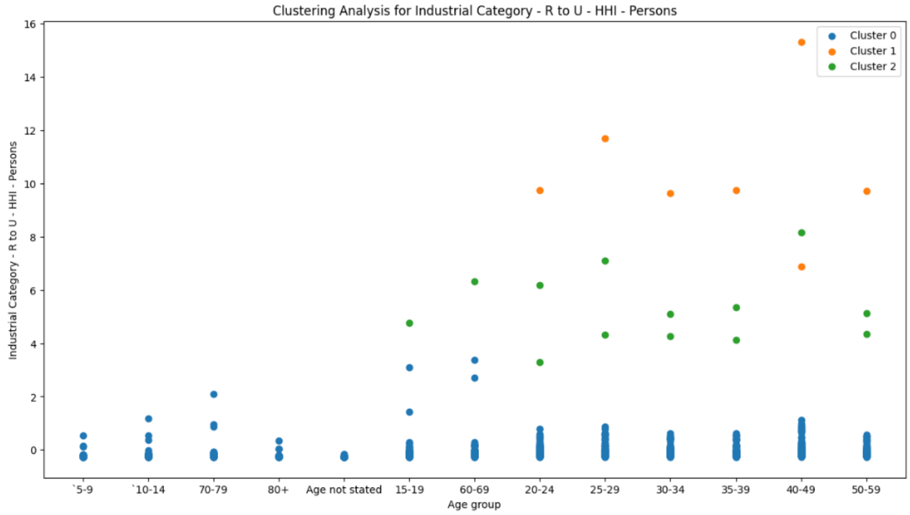


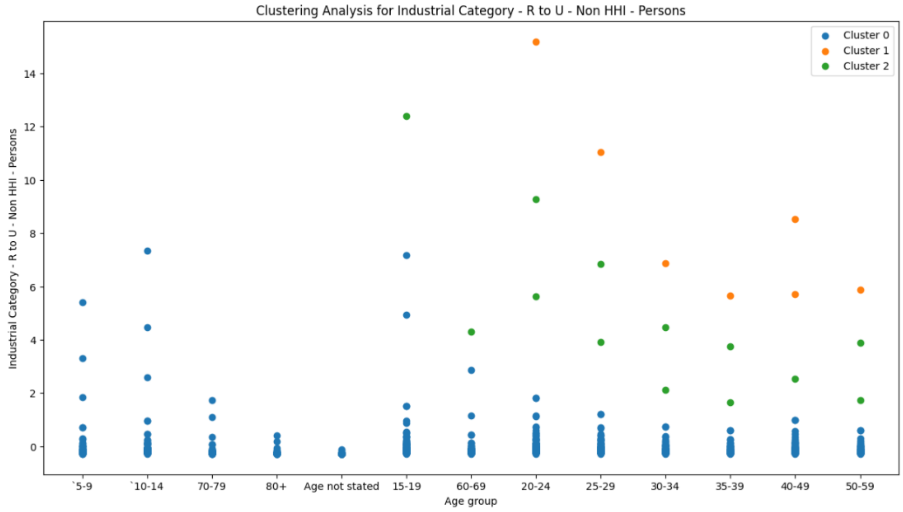
Clustering Analysis for Industrial Category - K to M - Persons




Clustering Analysis for Industrial Category - N to O - Persons
Industrial Category - N to O - Persons
Age group
Cluster 0
Cluster 1
Cluster 2
0
2
4
6
8
10
12
14
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59


Clustering Analysis for Industrial Category - P to Q - Persons
Industrial Category - P to Q - Persons
Age group
Cluster 0
Cluster 1
Cluster 2
0
2
4
6
8
10
12
14
16
`5-9
`10-14
70-79
80+
Age not stated
15-19
60-69
20-24
25-29
30-34
35-39
40-49
50-59

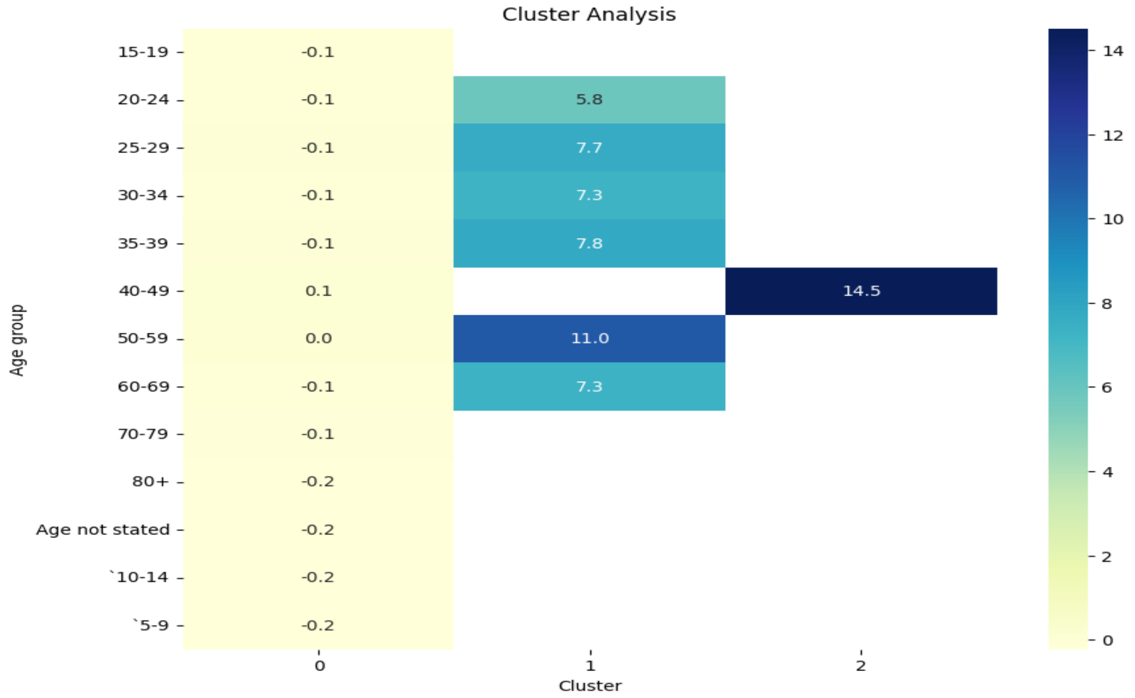




Heat Map for industrial category A(sample)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming df is your DataFrame after data cleaning and clustering#
Create a pivot table for better visualization
pivot_table = df.pivot_table(index='Age group', columns='Cluster',
values='Industrial Category - A - Cultivators - Persons',
aggfunc='mean')

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(pivot_table, annot=True, fmt=".1f", cmap='YlGnBu')
plt.title('Cluster Analysis')
plt.show()
```

Output:



Visualizations

Scatter plot and heat map was generated to visualize the clusters for different industrial categories.(Category A and B were gtaken as samples) Each plot displayed data points colored according to their respective cluster.

Interpretation of Clusters

The clusters revealed distinctive patterns among different industrial categories and age groups. Further analysis of each cluster provided valuable insights into the workforce distribution and trends.

Innovation in Design

An innovative aspect of this analysis was the combination of industrial category-specific metrics and age groups in the clustering process. This approach allowed for a more comprehensive understanding of workforce dynamics.

Recommendations

Based on the clustering results, the following recommendations are provided:

- Targeted skill development programs for specific industrial categories.
- Tailored workforce policies for different age groups within each category.
- Resource allocation strategies considering the identified clusters

Conclusion

The clustering analysis successfully identified meaningful patterns among industrial categories and age groups. It can be observed that the marginal workers in the age group 40-49 are more in strength compared to other age groups .

The insights derived from this analysis have the potential to significantly impact policy-making and resource allocation strategies.