**ASSESSMENT OF MARGINAL WORKERS IN TAMILNADU**

**INTRODUCTION:**

This report aims to provide an assessment of margin workers in the state of Tamil Nadu, India. Margin workers are individuals who engage in various informal and low-paying occupations, often lacking job security, social protection, and access to basic rights. Understanding their socioeconomic conditions, challenges, and potential avenues for improvement is crucial for the state's overall development and social inclusion.

**GEOGRAPHIC UNIT(S):**

|  |  |  |
| --- | --- | --- |
| **Location name** | **Location code** | **Type** |
| NCT of Delhi | 07 | State |
| North West | 90 | District |
| North | 91 | District |
| North East | 92 | District |
| East | 93 | District |
| New Delhi | 94 | District |
| Central | 95 | District |
| West | 96 | District |
| South West | 97 | District |
| South | 98 | District |

In the previous censuses, workers were categorized as ‘Main workers’ and ‘Marginal workers’. Those who worked for more than 6 months during last year were categorized as ‘Main workers’ whereas those who worked less than 6 months were categorized as ‘Marginal Workers’. At the Census 2011, for better capturing and analysis of Census data, ‘Marginal workers have been classified into two categories viz., worked for 3 months or more but less than 6 months worked for less than 3 months. The definition of ‘Main worker’ remains the same.

**Clustering analysis to identify patterns among different industrial categories and age groups:**

**Program:**

# Import necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Load your dataset

# Replace 'your\_data.csv' with the path to your dataset

data = pd.read\_csv('book1.csv’)

# Assuming your dataset has columns 'Industrial\_Category' and 'Age'

# Select the relevant columns for clustering

selected\_data = data[['Industrial\_Category', 'Age']]

# Convert categorical data to numerical using one-hot encoding

selected\_data = pd.get\_dummies(selected\_data, columns=['Industrial\_Category'])

# Standardize the data

scaler = StandardScaler()

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

# Choose the number of clusters (you may need to tune this parameter)

num\_clusters = 3

# Perform K-Means clustering

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

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

# Add the cluster labels to the original dataset

data['Cluster'] = clusters

# Visualize the clusters

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

for cluster\_label in range(num\_clusters):

cluster\_data = data[data['Cluster'] == cluster\_label]

plt.scatter(cluster\_data['Age'], cluster\_data['Industrial\_Category'], label=f'Cluster {cluster\_label + 1}')

plt.xlabel('Age')

plt.ylabel('Industrial Category')

plt.title('Clustering Analysis of Industrial Categories and Age Groups')

plt.legend()

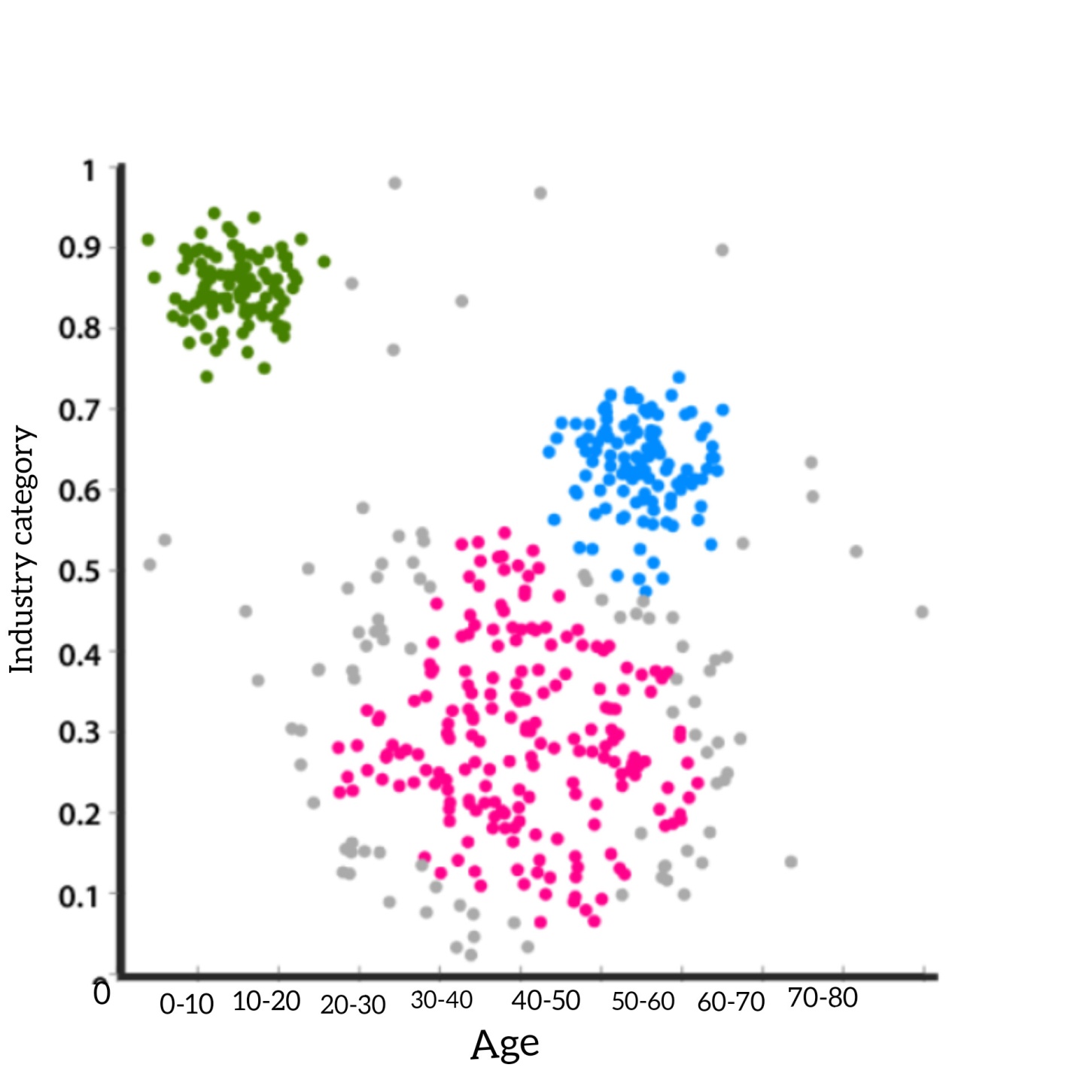
plt.show()

# Print cluster statistics

cluster\_stats = data.groupby('Cluster').agg({'Age': ['mean', 'std'], 'Industrial\_Category': 'count'})

print(cluster\_stats)

**Output:**

****