#### Problem statement:

Analyze the demographic characteristics of marginal workers based on age, industrial category, and sex. Create visualizations such as bar charts, pie charts, or heatmaps to represent the distribution across different categories.

**Demographic characteristics:**

An analysis of the demographic characteristics of marginal workers begins by collecting data on three important factors: age, industry, and gender.

**Data collection:** Collect data needed for analysis.

**Data cleaning and preparation:** Clean and prepare data for analysis.

**Data Analysis:** Use statistical methods to analyze data and identify trends and patterns.

**Visualization:** Create visualization to show the distribution of marginal workers in different categories.

**1. Age distribution:**

To analyze the age distribution of marginal workers, you can collect data on the age groups of marginal workers and create bar charts or histograms to display this information. Here's how:

* Collect data on age groups of marginal workers (eg 18-24, 25-34, 35-44, etc.).
* Count the number of marginal workers in each age group.
* Create a bar graph with age groups on the x-axis and the marginal number of workers on the y-axis.

This bar graph shows the distribution of marginal workers in different age groups.

**2. Industry category:**

To analyze the distribution of marginal workers by industry classification, follow these steps:

* Collecting data on different categories of industry (agriculture, manufacturing, services, etc.).
* Count the number of marginal workers by industry classification.
* Create a bar or pie chart to show the distribution of marginal workers in different industry categories.

A bar chart lets you compare the number of marginal workers in each category, while a pie chart shows the percentage of marginal workers in each category relative to the total.

**3. Gender distribution:**

To analyze the distribution of marginal workers by gender, follow the steps below:

* Collecting data on the gender of marginal workers (male, female, other).
* Count the number of marginal workers by gender classification.
* Create a pie chart or bar chart to show the distribution of marginal workers by gender.

The pie chart visually shows the percentage of marginal workers in each gender category and the bar chart

**Data Collection:**

Begin by gathering data. You should have a dataset that includes information on industrial categories (e.g., types of industries) and age groups. Each data point should be associated with both a category and an age group. Collected data from the given project to analysis the categories of the people distributed in various districts, ages ,working hours ,industry categories

**Data Preprocessing:**

● Ensure that data is clean and free from errors or missing values

● Standardize or normalize data if necessary to bring all variables to a common scale.

● Null value, age not stated these values are optimised

1. Choosing a Clustering Algorithm:

● Selecting an appropriate clustering algorithm.

Two common choices are:

● K-Means Clustering: Divides data into 'k' clusters based on similarity.

● Hierarchical Clustering: Forms a tree-like structure of clusters, showing relationships at different levels of granularity.

1. Feature Selection and Engineering:

● Decide which features (variables) you want to use for clustering. In case, it's industrial categories and age groups.

● You might need to transform or combine features to make them suitable for clustering.

**Clustering Analysis:**

● Apply the chosen clustering algorithm to data.

● For K-Means, you'll need to specify the number of clusters (k) in advance.

● For Hierarchical Clustering, you can choose to cut the tree at a certain level to define clusters.

**Interpretation and Visualization:**

● Visualize the clusters using plots or graphs to understand how data points are grouped.

● Explore the characteristics of each cluster to interpret the patterns that emerge.

**Validation and Refinement:**

● Evaluate the quality of clusters using appropriate metrics (e.g., silhouette score, Davies-Bouldin index).

● If the results are not satisfactory, consider adjusting parameters or trying different algorithms.

**Insights and Recommendations:**

● Based on clustering results, draw insights and make recommendations. For example, you might discover that certain age groups are more prevalent in specific industries, which can inform marketing or policy decisions.

**Documentation and Reporting:**

● Documenting entire process, including data preprocessing, clustering parameters, and results.

● Presenting findings in a clear and understandable manner, possibly through reports or presentations.

**Iterate if Needed:**

● Depending on goals, you may need to refine analysis or perform additional iterations to gain deeper

**Data preprocessing and importing**

To start the data analysis process by loading and preprocessing a dataset using Python, you'll typically use the “pandas” library. I'll walk you through the steps to do this assuming you have a dataset in a CSV file. Make sure you have `pandas` installed. If not, you can install it using”pip”:

Code to install panadas

pip install pandas

Here's how to load and preprocess a dataset:

**1.** **Import the necessary libraries:**

Program to import library

import pandas as pd

**2. Load the dataset using “pandas”:**

Program to load the data

dataset = pd.read\_csv('tn\_age\_distribution\_dataset.csv')

**3. Explore the dataset:**

* To get a quick overview of the dataset, you can use “head()”to display the first few rows of the data:

Code to get first five rows

print(dataset.head())

* To get a summary of the dataset's statistics, you can use “describe()”:

Program to get summary of the datas:

print(dataset.describe())

* To check for missing values, you can use “isnull()” and “sum()”:

Program to remove the null values

print(dataset.isnull().sum())

**4. Preprocess the dataset:**

* Handle missing values: Depending on the dataset, you might want to fill missing values with appropriate values or drop rows or columns with missing data.
* Data type conversion: Ensure that the data types of columns are appropriate. You can use “astype()”to convert data types.
* Remove duplicates: Use “drop\_duplicates”` to remove duplicate rows, if necessary.
* Feature engineering: Create new columns, modify existing ones, or encode categorical variables as needed.
* Normalize or scale numerical features, if required.

**5. Save the preprocessed dataset (optional):**

If you want to save the preprocessed dataset for future use, you can use “to\_csv()”:

Program to save cleaned data:

# Replace “tn\_age\_ditribution\_preprocessed\_dataset.csv”

your desired file name

dataset.to\_csv(“tn\_age\_ditribution\_preprocessed\_dataset.csv”, index=False)

def personsUrbanOrR

# to draw the graph of ratio between the rural and urban in each industries according ural(df):

    cols = df.columns

    base = ["Area Name","Total/ Rural/ Urban","Age group"]

    persons = [item for item in cols if item.endswith("Persons")]

    persons = base + persons

    persons = df[persons]

    persons = persons[persons["Age group"] == "Total" ]

    persons = persons[persons["Area Name"] != "State - TAMIL NADU"]

    persons = persons[persons["Total/ Rural/ Urban"] != "Total"]

    persons= persons.drop("Age group",axis=1)

    persons = persons.reset\_index()

    persons = persons.drop("index",axis=1)

    persons

    industry = persons.columns

    print(industry)

    for i in range(2,len(industry)):

        pivot\_data = persons.pivot(index='Area Name', columns='Total/ Rural/ Urban', values=[industry[i]])

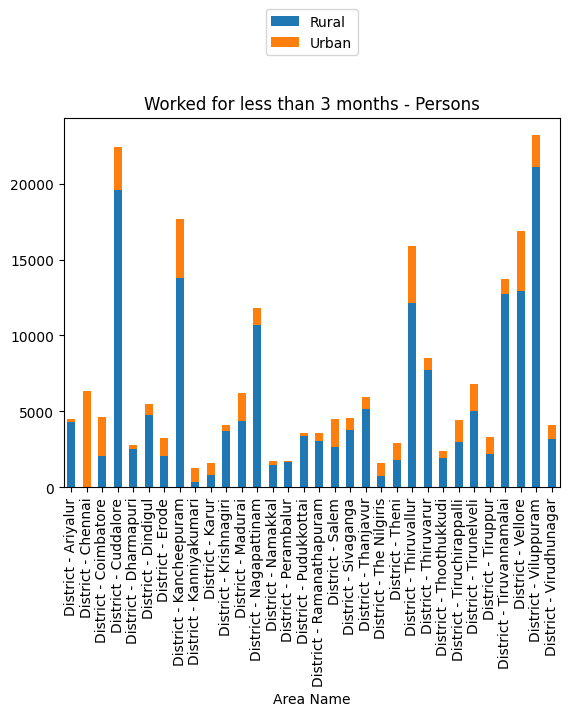
        ax = pivot\_data.plot(kind="bar",stacked=True,title=industry[i])

        legand\_label = ["Rural","Urban"]

        ax.legend(labels=legand\_label,loc='lower center', bbox\_to\_anchor=(0.5, 1.15))

        plt.show()

personsUrbanOrRural(df)



#To find the total workers no of person worked in each district

def personsTotal(df):

    cols = df.columns

    base = ["Area Name","Total/ Rural/ Urban","Age group"]

    persons = [item for item in cols if item.endswith("Persons")]

    persons = base + persons

    persons = df[persons]

    persons = persons[persons["Age group"] == "Total" ]

    persons = persons[persons["Total/ Rural/ Urban"] == "Total"]

    persons= persons.drop("Age group",axis=1)

    persons = persons.drop("Total/ Rural/ Urban",axis=1)

    persons = persons.drop(0)

    persons = persons.reset\_index()

    persons = persons.drop("index",axis=1)

    title=persons.columns

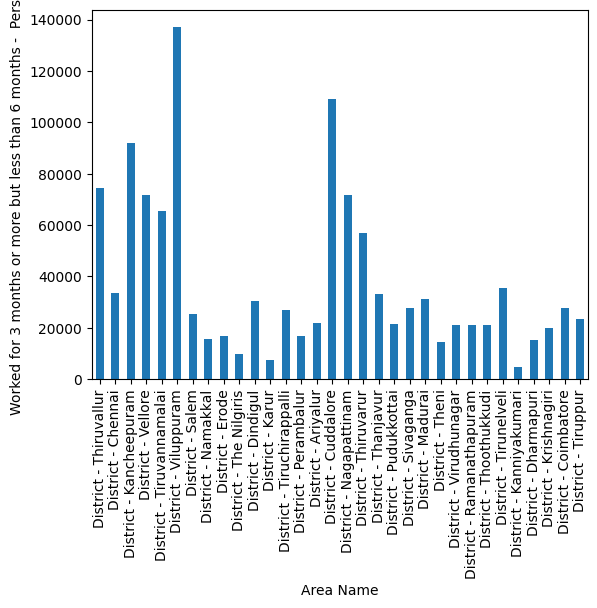
    for i in  range(1,len(title)):

        persons.plot(x="Area Name", y=title[i], kind='bar',legend=False)

        plt.ylabel(title[i])

        plt.show()

personsTotal(df)



#to analysis the age group by industrydata ,district,category

def AnalysisByIndustry(industryData,district,category):

    district = industryData[industryData["Area Name"]==district]

    district = district.drop("Area Name", axis=1)

    district = district[district["Total/ Rural/ Urban"]==category]

    district = district.drop("Total/ Rural/ Urban",axis=1)

    district = district.drop(district.index[0])

    district.plot(x="Age group",kind="bar",stacked=True)

    plt.show()

AnalysisByIndustry(industriesName[1],"District - Tiruppur","Total")

