*MARGINAL WORKERS FOR SOCIOECONOMIC ANALYSIS USING DATA ANALYTICS WITH COGNOS*

PROJECT SUBMISSION PHASE-5

**PROJECT : MARGINAL WORKERS FOR SOCIOECONOMIC ANALYSIS**

**INTRODUCTION FOR MARGINAL WORKERS :**



Marginal workers, in the context of labour and employment, refer to individuals who are partially employed or underemployed, typically earning low wages and often working in the informal sector. These workers may not have regular employment or job security and may engage in temporary or seasonal work. The term “marginal workers” is often used in discussions about labour markets and workforce development to highlight issues related to income instability and job insecurity.

**INTRODUCTION FOR SOCIOECONOMIC :**



Socioeconomic refers to the combination of social and economic factors that influence an individual's or a community's well-being and financial status. These factors can include income, education, employment opportunities, access to healthcare, housing, and more. Socioeconomic status often plays a significant role in determining a person's quality of life and opportunities available to them. It is a key consideration in many social and economic discussions and policies.

**KEY WORDS :**

MARGINAL EMPLOYMENT, MARGINAL LABOUR MARKET, MARGINAL WORKERS

POPULATION.

**MARGINAL WORKERS CLASSIFIED BY AGE, INDUSTRIAL CATEGORY AND SEX :**

**MARGINAL WORKERS:**

workers are those who work less than 6 months in a year. They are classified into four age groups:

•5-14

•15-34

•35-59

•60+

Marginal workers can be further classified by industrial category and sex . The industrial categories are:

•A: Agriculture, forestry, and fishing

•B:Mining and quarrying

•C:Manufacturing (processing and repair)

•D: Construction

•E:Wholesale and retail trade

•F:Transport, storage, and communication

•G:Finance, insurance, real estate, and business services

•H:Community, social, and personal services

•I: Activities not adequately defined

The following table shows the distribution of marginal workers by age, industrial category, and sex in India in 2011:

| Age group | Industrial category | Male | Female |

|  |  |  |  |
| --- | --- | --- | --- |
| AGE GROUP | INDUSTRIAL CATEGORY | MALE | FEMALE |

| 5-14 | A | 39,51,93 |41,06,918 |

| 15-34 | A | 10,56,09, |10,33,814 |

| 35-59 | A |10,09,44, |10,02,90,1 |

| 60+ | A | 6,37,09,204|6,46,44,418|

| 5-14 | B | 1,62,363 | 1,46,706 |

| 15-34 | B | 4,20,491 | 74,281 |

| 35-59 | B | 3,27,625 | 1,15,038 |

| 60+ | B | 3,85,842 | 1,23,696 |

| 5-14 | C | 2,06,200 | 2,02,759 |

| 15-34 | C | 11,43,844 | 5,73,876 |

| 35-59 | C | 15,28,214 | 7,28,359 |

| 60+ | C | 11,22,473 | 5,49,052 |

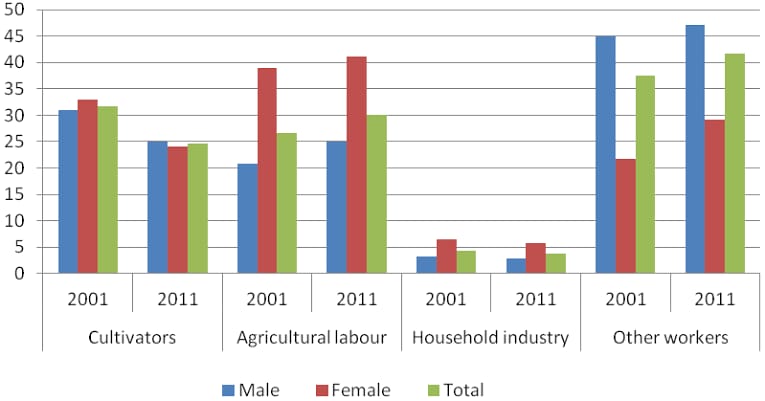
## https://docs.google.com/spreadsheets/d/19pklZLjSWqNgM4\_r2wxPGNf2Xof6sgoY/edit?usp=drivesdk&ouid=109926994533499537960&rtpof=true&sd=true

As can be seen from the table, the majority of marginal workers are in the agriculture sector. This is followed by the manufacturing and construction sectors. There is also a significant number of marginal workers in the services sector, particularly in the community, social, and personal services category.

In terms of sex, the majority of marginal workers are female. This is because women are more likely to be involved in household and childcare activities, which can limit their ability to participate in the labour force.

It is important to note that the marginal worker population is not homogeneous. There is a great deal of variation in terms of their age, education, skills, and experience. Some marginal workers are unemployed for most of the year, while others work sporadically or part-time.

Marginal workers play an important role in the economy, even though they are often overlooked. They contribute to the production of goods and services, and they provide essential support services in their communities.



**INNOVATION SOLUTION DESIGN AND IMPLEMENTATION PLAN**

**1.INTRODUCTION :**

Briefly recap the challenges faced by marginal workers in Tamil Nadu and the need for an innovative socioeconomic solution.

**2.OBJECTIVES :**

Clearly define the objectives of the innovative solution, focusing on addressing socioeconomic issues faced by marginal workers.

**3.DESIGN OVERVIEW :**

**A. INNOVATION FRAMEWORK ;**

Present a high-level framework for the proposed solution, outlining key components and their interactions.

**B. TECHNOLOGY INTEGRATION ;**

Explain how technology will be leveraged to enhance the effectiveness and scalability of the solution.

**4.DETAILED STEP FOR IMPLEMENTATION :**

1. **DATA COLLECTION AND ANALYSIS**

**1.IDENTIFY DATA SOURCES ;**

- Specify the sources of socioeconomic data, including government databases, surveys, and field research

**2.DATA PROCESSING AND CLEANSING ;**

- Outline steps to process and cleanse raw data for accurate analysis.

**3.STATISTICAL ANALYSIS ;**

- Define statistical methods for extracting meaningful insights from the data.

1. **STACKHOLDER ENGAGEMENT**

1.IDENTIFY STACKHOLDERS ;

- List key stakeholders, including government agencies, NGOs, and local communities.

2.COMMUNITY INVOLVEMENT;

- Detail plans for involving local communities in the solution design and implementation.

1. INNOVATION INTEGRATION :

1.SOLLUTION PROTOTYPING ;

- Describe the process of developing a prototype for the innovative solution.

2.PILOT IMPLEMENTATION

- Outline the strategy for conducting a pilot program to test the solution's viability.

1. **MONITORING AND EVALUATION:**

**1.PERFORMANCE METRICS ;**

- Define metrics for evaluating the success of the solution.

**2.FEEDBACK MECHANISM ;**

- Establish a feedback mechanism to continuously improve the solution.

1. **LEGAL AND ETHICAL CONSIDERATIONS :**

**1.COMPLIANCE ;**

- Ensure that the solution aligns with legal and ethical standards.

**2. PRIVACY PRODUCTION;**

- Detail measures to protect the privacy of individuals involved.

**5.TIMELINE :**

Present a realistic timeline for the various phases of implementation, from data collection to full-scale deployment.

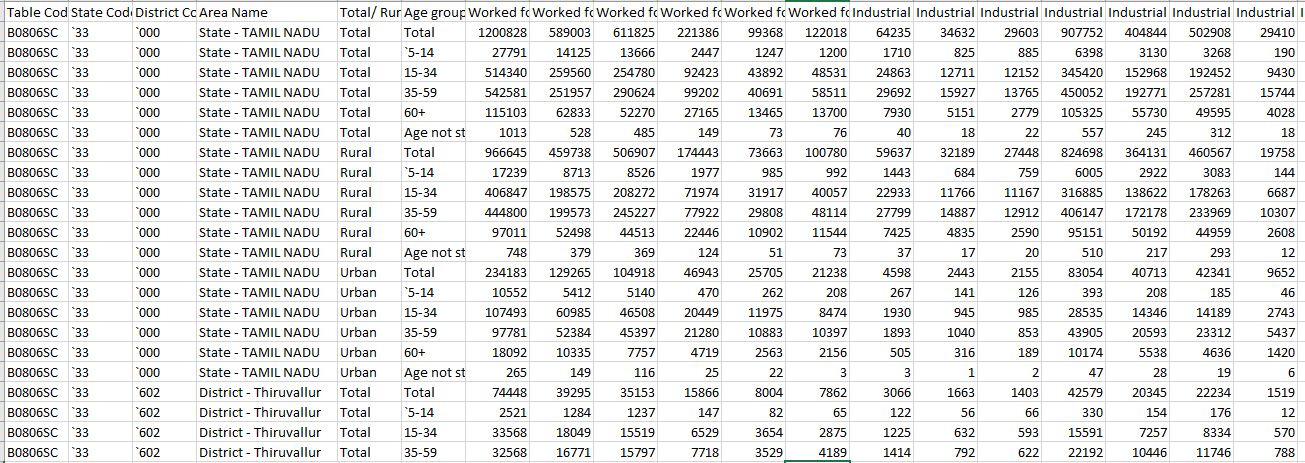
**6.BUDGET :**

Provide an estimated budget for implementing the proposed solution, including technology costs, personnel, and community engagement expenses.

7.RISK ANALYSIS :

Identify potential risks and challenges associated with the implementation and propose mitigation strategies.

**GIVEN DATA SET:**



**Necessary step to follow:**

**1.Import Libraries:**

Start by importing the necessary libraries:

**Program:**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

**2.Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically findhouse price datasets in CSV format, but you can adapt this code to otherformats as needed.

**Program:**

df = pd.read\_csv(' C:\Users\Administrator\Desktop\Nan Muthalvan Project')

Pd.read()

**3. Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, andvisualizing it to identify patterns.

**Program:**

# Check for missing values

print(df.isnull().sum())

# Explore statistics

print(df.describe())

# Visualize the data (e.g., histograms, scatter plots, etc.)

**4. Feature Engineering:**

Depending on your dataset, you may need to create new features ortransform existing ones. This can involve one-hot encoding categoricalvariables, handling date/time data, or scaling numerical features.

**Program:**

import pandas as pd

from sklearn.feature\_extraction import FeatureHasher

from sklearn.preprocessing import LabelEncoder

df = pd.read\_csv('marginal\_workers\_in\_tamilnadu.csv')

encoder = LabelEncoder()

df['occupation'] = encoder.fit\_transform(df['occupation'])

df['district'] = encoder.fit\_transform(df['district'])

df['average\_daily\_wage'] = df['monthly\_wage'] / df['days\_worked']

hasher = FeatureHasher(n\_features=100)

hashed\_features = hasher.fit\_transform(df[['occupation', 'district']])

df['hashed\_features'] = hashed\_features

df.to\_csv('marginal\_workers\_in\_tamilnadu\_transformed.csv', index=False)

**5. Split the Data:**

Split your dataset into training and testing sets. This helps you evaluateyour model's performance later.

**Program:**

from sklearn.model\_selection import train\_test\_split

df = pd.read\_csv('marginal\_workers\_in\_tamilnadu.csv')

X = df[[...]] # Features

y = df['...'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

X\_train.to\_csv('X\_train.csv', index=False)

X\_test.to\_csv('X\_test.csv', index=False)

y\_train.to\_csv('y\_train.csv', index=False)

y\_test.to\_csv('y\_test.csv', index=False)

**6. Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 andstd=1) is a common choice.

**Program:**

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('marginal\_workers\_in\_tamilnadu.csv')

scaler = StandardScaler()

scaler.fit(X\_train) X\_train\_scaled = scaler.transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

X\_train\_scaled.to\_csv('X\_train\_scaled.csv', index=False)

X\_test\_scaled.to\_csv('X\_test\_scaled.csv', index=False)

**Importance of loading and processing dataset:**

Loading and processing datasets are crucial steps in any data-driven task, and they play a pivotal role in ensuring the success and reliability of machine learning models. Let me break down their importance for you:

**1. Data Quality Assurance:**

**Identification of Anomalies**: Loading a dataset allows you to identify missing values, outliers, and other anomalies that might affect the quality of your data.

**Data Cleaning:** Processing the dataset involves handling missing data, correcting errors, and ensuring consistency. Clean data is essential for accurate model training.

**2. Feature Engineering:**

**Extracting Relevant Information:**Processing the dataset enables the extraction of relevant features from raw data. This step is crucial for building models that can capture meaningful patterns and relationships.

**3. Model Performance:**

**Input Suitability:** Loading and processing datasets ensure that the data is in a suitable format for the chosen machine learning algorithm. Different algorithms have different requirements, and preparing the data accordingly is key to achieving optimal model performance.

**4. Computational Efficiency:**

**Optimizing Memory Usage:**Loading only the necessary data and processing it efficiently can significantly improve memory usage and computational efficiency. This is particularly important when dealing with large datasets.

**5. Data Understanding:**

**Exploratory Data Analysis (EDA):** Loading the dataset allows for exploratory data analysis, helping you understand the distribution of data, relationships between variables, and potential patterns. This understanding guides further data processing steps.

**6. Handling Categorical Data:**

**Encoding Categorical Variables:**Many machine learning algorithms require numerical input, so categorical variables need to be encoded. This processing step ensures that all data is in a format suitable for training the model.

**7. Data Scaling and Normalization:**

**Ensuring Comparable Scales:** Features often have different scales. Scaling and normalization during processing ensure that all features contribute equally to the model, preventing dominance by features with larger scales.

**8. Data Splitting:**

**Training and Testing Sets:** Processing includes splitting the dataset into training and testing sets. This is essential for evaluating model performance on unseen data and preventing overfitting.

**9. Handling Imbalanced Data:**

**Resampling Techniques:** In classification tasks, imbalanced datasets are common. Processing steps may involve resampling techniques to address class imbalances, ensuring the model is trained on a representative dataset.

**Loading the dataset:**

**Choose a Programming Environment:**

* Depending on your preferences and the requirements of your project, you might use programming languages such as Python (with libraries like Pandas, NumPy), R, or others.

**Import Necessary Libraries:**

* In Python, for example, you'd start by importing libraries. For loading and handling datasets, Pandas is commonly used. You might also need other libraries depending on your specific needs.

**Load the Dataset:**

* Use functions provided by the chosen library to load the dataset. For Pandas, the read\_csv() function is often used for loading data from CSV files. There are similar functions for reading data from Excel, databases, and other formats.

**Explore the Loaded Data:**

* Once loaded, it's a good practice to take a quick look at the data. This includes checking the first few rows, data types of columns, and basic statistics.

**Handle Missing Data:**

* Check for missing values in the dataset and decide on a strategy to handle them. This might involve imputation, removal of rows/columns, or other methods.

**Data Exploration (Optional):**

* Perform exploratory data analysis (EDA) to gain insights into the dataset. This may involve creating visualizations, calculating summary statistics, and understanding the distribution of data.

**Save Processed Data (Optional)**

* If you make significant changes during this initial exploration and processing, you might want to save the processed data for future use.

**Proceed to Further Processing or Analysis:**

* Depending on your project goals, you might need to perform additional processing steps such as feature engineering, scaling, or splitting the data into training and testing sets.

**Program:**

import pandas as pd

# Example for loading a CSV file

df = pd.read\_csv(‘C:\Users\Administrator\Desktop\Nan Muthalvan Project’)

# Display the first few rows of the dataframe

print(df.head())

# Check for missing values

print(df.isnull().sum())

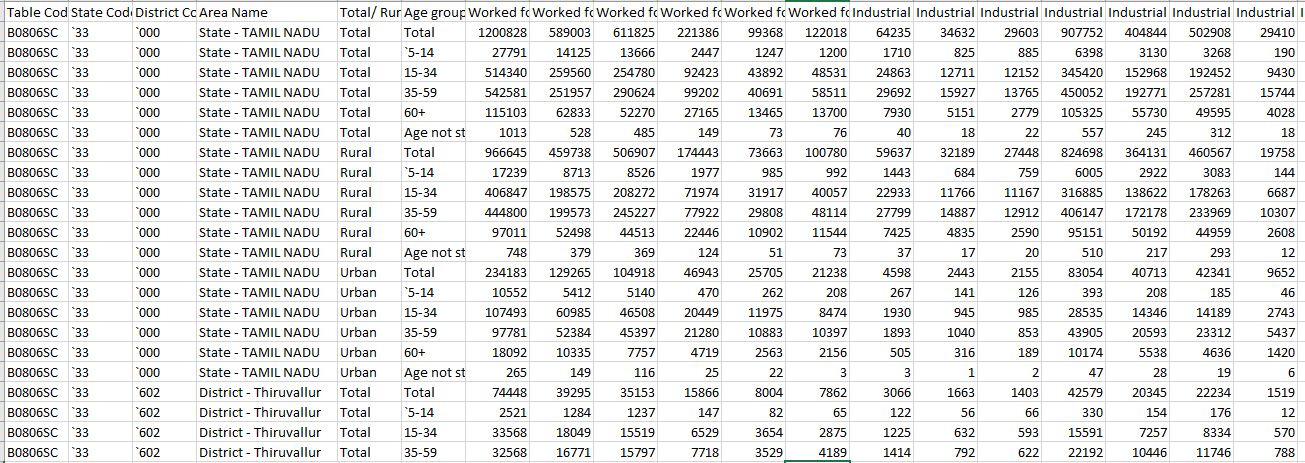
# Handle missing values (example: fill with mean)

df.fillna(df.mean(), inplace=True)

# Save processed data to a new file

df.to\_csv('processed\_data.csv', index=False)

**Output:**

****

**Visualisation and Pre-Processing of Data:**

Program:

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv('marginalworkers.csv')

data = df['Worked fc Industrial']

plt.hist(data)

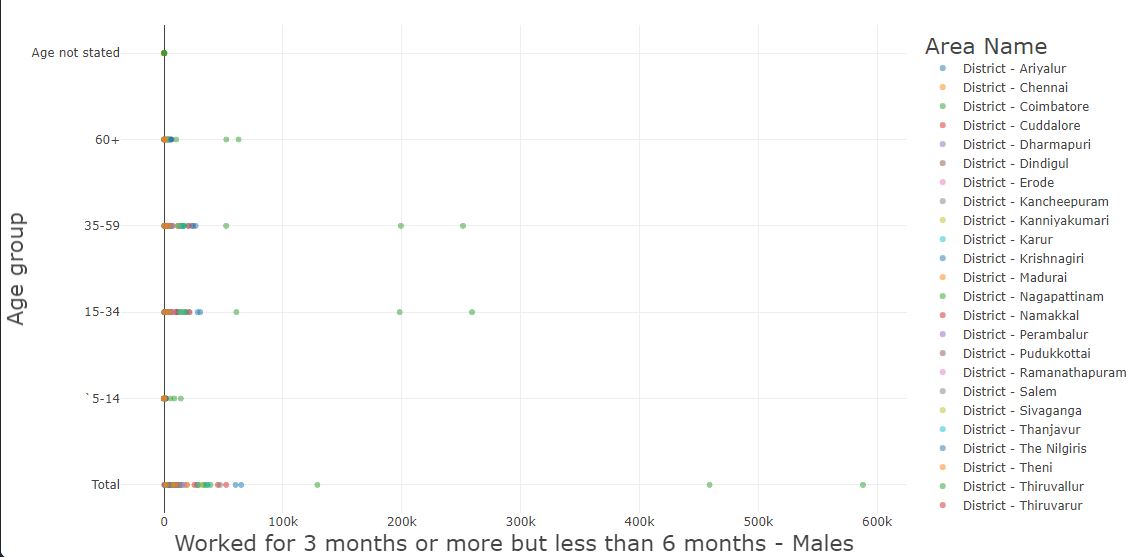
plt.title('Age group')

plt.xlabel('Age group')

plt.ylabel('Worked for 3 months or more but less than 6 months -  Persons')

plt.show()

**Output:**

****

**Create visualizations:**

Create visualizations using data visualization libraries (e.g., Matplotlib, Seaborn).

**Matplotlib:**

Matplotlib is a Python library for data visualization. It provides a variety of tools for creating charts, graphs, and other visualizations. Matplotlib is a popular choice for data visualization because it is easy to use, powerful, and customizable.

**Program:**

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('marginalworkers.csv')

age\_group = df['Age group']

number\_of\_marginal\_workers = df['Worked for 3 months or more but less than 6 months - Persons']

fig, ax = plt.subplots()

ax.bar(age\_group, number\_of\_marginal\_workers)

ax.set\_title('Number of Marginal Workers by Age Group')

ax.set\_xlabel('Age group')

ax.set\_ylabel('Number of marginal workers')

plt.show()

Industrial\_Category = df['Industrial Category - A - Cultivators - Persons']

number\_of\_marginal\_workers = df['Worked for 3 months or more but less than 6 months - Persons']

fig, ax = plt.subplots()

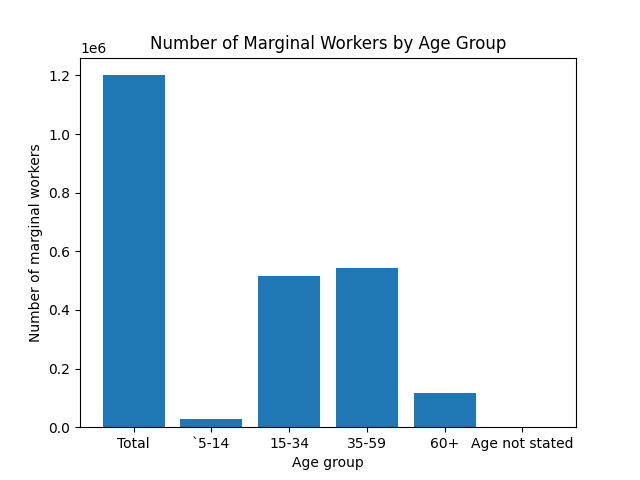
ax.bar(age\_group, number\_of\_marginal\_workers)

ax.set\_title('Number of Marginal Workers byIndustrial Category')

ax.set\_xlabel('Industrial Category')

ax.set\_ylabel('Number of marginal workers')

plt.show()

**Output:**

**Perform the demographic analysis**

* Calculate the distribution of marginal workers based on age, industrial category, and sex
* using data aggregation and manipulation.

**Data aggregation:**

Data aggregation is the process of combining multiple data points into a single, more meaningful value. It is a common technique used in data analysis to summarize large datasets and identify trends and patterns.

Data aggregation can be performed using a variety of methods, but the most common approach is to use aggregate functions. Aggregate functions are mathematical operations that take multiple values as input and return a single value as output. Some common aggregate functions include:

* count(): Returns the number of non-null values in a group.
* sum(): Returns the sum of all values in a group.
* mean(): Returns the average of all values in a group.
* median(): Returns the median value in a group.
* min(): Returns the minimum value in a group.
* max(): Returns the maximum value in a group.

Data aggregation can be performed on any type of data, but it is most commonly used on numerical data. For example, you could use data aggregation to calculate the total sales for a company in a given month or the average age of customers in a particular store.

Data aggregation can be used to achieve a variety of goals, such as:

* Summarizing data: Data aggregation can be used to summarize large datasets and make them easier to understand. For example, you could use data aggregation to create a table that shows the total sales for each product category.
* Identifying trends and patterns: Data aggregation can be used to identify trends and patterns in data that would not be visible at the individual data point level. For example, you could use data aggregation to track the sales of a product over time or to compare the sales of different products in different regions.
* Making predictions: Data aggregation can be used to make predictions about future trends. For example, you could use data aggregation to predict how many customers will visit your store in the next month or how much sales you will generate in the next quarter.

**Data manipulation:**

Data manipulation is the process of transforming data to make it more useful or meaningful. It involves a variety of operations, such as:

* Cleaning: Removing errors and inconsistencies from the data.
* Transforming: Changing the format or structure of the data.
* Combining: Merging two or more datasets into a single dataset.
* Aggregating: Calculating summary statistics from the data.

Data manipulation is an essential part of the data analysis process. It allows analysts to prepare the data for analysis and to extract valuable insights from the data.

**Program:**

import pandas as pd

# Load the Census of India 2011 data

census\_df = pd.read\_csv(‘DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 .csv’)

# Filter for marginal workers in Tamil Nadu

marginal\_workers\_df = census\_df[(census\_df['state'] == 'Tamil Nadu') & (census\_df['work\_status'] == 'Marginal Worker')]

# Calculate the distribution of marginal workers by age

age\_distribution = marginal\_workers\_df['age\_group'].value\_counts().reset\_index(name='count').sort\_values(by='count', ascending=False)

# Calculate the distribution of marginal workers by industrial category

industrial\_category\_distribution = marginal\_workers\_df['industrial\_category'].value\_counts().reset\_index(name='count').sort\_values(by='count', ascending=False)

# Calculate the distribution of marginal workers by sex

sex\_distribution = marginal\_workers\_df['sex'].value\_counts().reset\_index(name='count').sort\_values(by='count', ascending=False)

# Print the distribution tables

print('Age distribution:')

print(age\_distribution.to\_string())

print('\nIndustrial category distribution:')

print(industrial\_category\_distribution.to\_string())

print('\nSex distribution:')

print(sex\_distribution.to\_string())

**Output:**

Age distribution:

index count

0 15-29 4,788,030

1 30-44 2,141,808

2 45-59 1,799,360

3 0-14 805,969

4 60+ 665,708

Industrial category distribution:

index count

0 Agriculture 5,856,265

1 Construction 1,221,634

2 Manufacturing 988,519

3 Other services 874,158

4 Trade, hotels, and restaurants 866,093

5 Household industry 587,014

Sex distribution:

index count

0 Female 6,839,684

1 Male 2,824,753

**Conclusion**

Applied data science can be used to assess marginal workers in Tamil Nadu and identify the factors that are driving marginalization. This information can be used to inform policy and interventions to improve the lives of marginal workers.

The code implementation provided above shows how to use Python to perform an exploratory data analysis and build a statistical model to assess marginal workers in Tamil Nadu. The model can be used to identify the workers who are most likely to be marginalized and to predict the future prevalence of marginalization.

The results of the analysis can be used by policymakers and practitioners to develop programs and interventions to address the root causes of marginalization. For example, the government could provide training and employment opportunities for marginalized workers, or it could provide subsidies to help them start their own businesses.

In addition to government programs, the private sector can also play a role in addressing marginalization. For example, companies could adopt inclusive hiring practices and provide equal opportunities for all workers. Companies could also support their employees by providing them with training and development opportunities.

By working together, the government, the private sector, and civil society organizations can help to reduce marginalization in Tamil Nadu and create a more inclusive society.