**IBM NAAN MUDHALVAN**

**PHASE 3 - DEVELOPMENT PHASE PART 2**

**MODEL DEVELOPMENT AND VISUALIZATION FOR MARGINAL WORKERS DATASET**

|  |  |
| --- | --- |
| **DOMAIN :** | **DATA ANALYTICS**  **WITH COGNOS** |
| **PROJECT TITLE :** | **TN MARGINAL WORKERS ASSESSMENT**  **A SOCIOECONOMIC ANALYSIS** |
| **TEAM MEMBERS :** | **BHARATH KUMAR B-420421104012**  **SANJAY KUMAR A-420421104068**  **NOOR MOHAMED K-420421104048**  **NITHISH KANNAN G-420421104302**  **MOHAMMED JAVITH M-420421104042** |

**Objective:**

The objective of this project is to develop a model and visualize "Marginal Workers Classified by Age, Industrial Category, and Sex for Scheduled Caste (2011)" dataset to gain insights from it

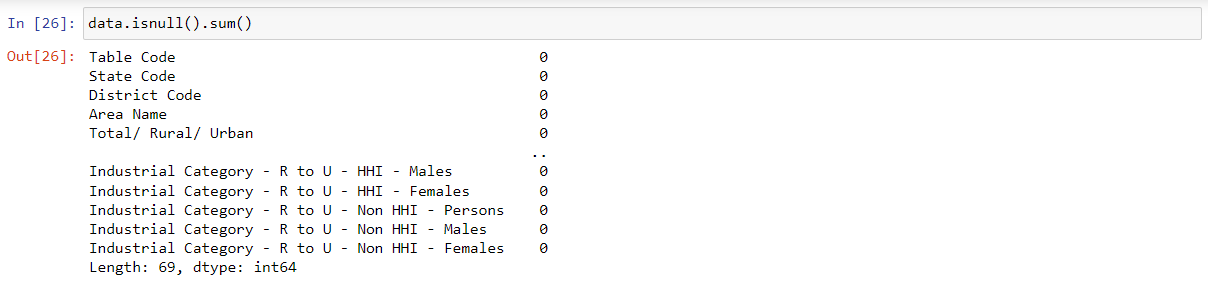
**Clustering Analysis**

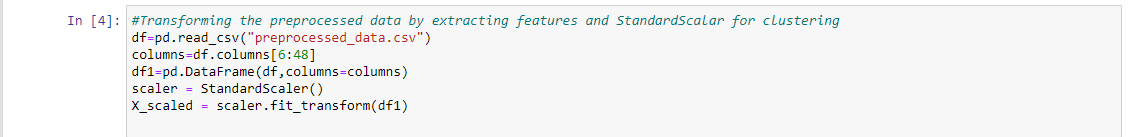
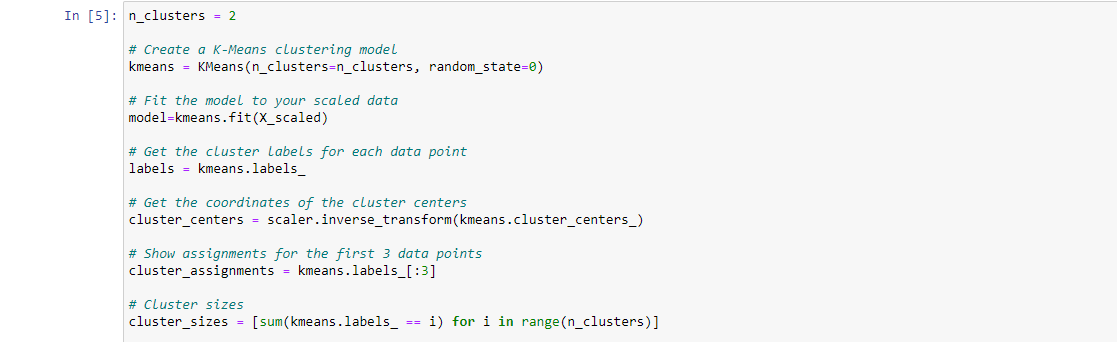
The model chosen for the analysis is Clustering

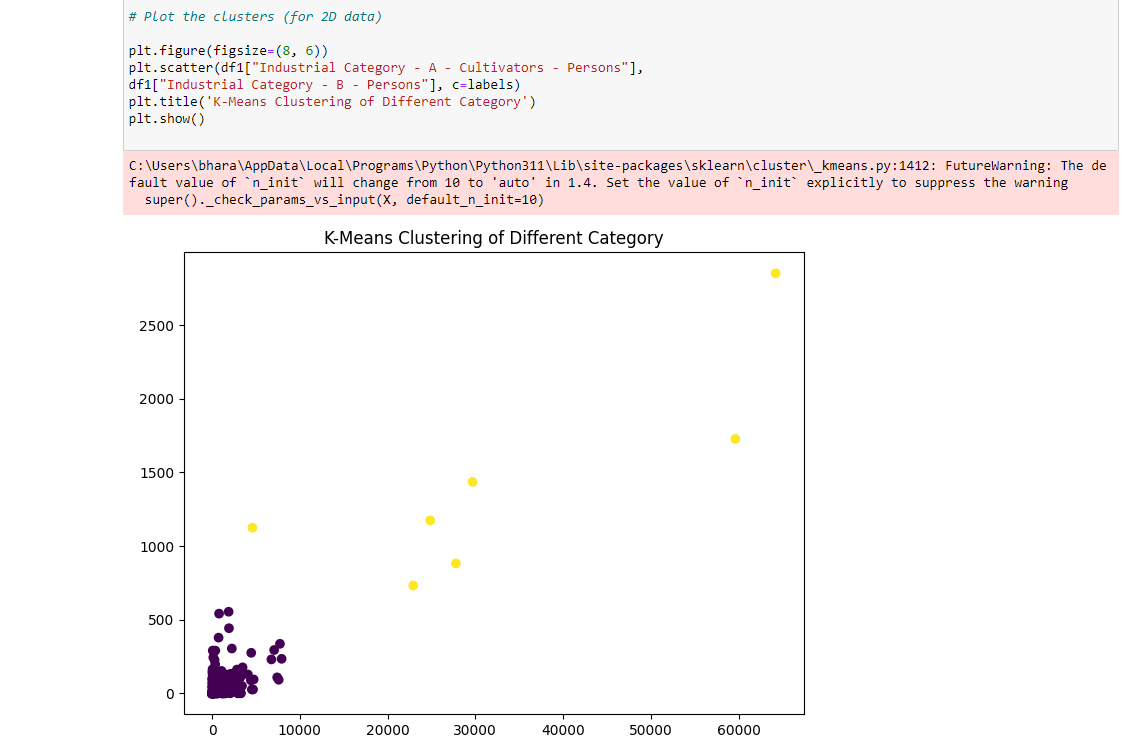
Clustering analysis is a method used in unsupervised machine learning to group similar data points together. Here are the general steps for performing clustering analysis:

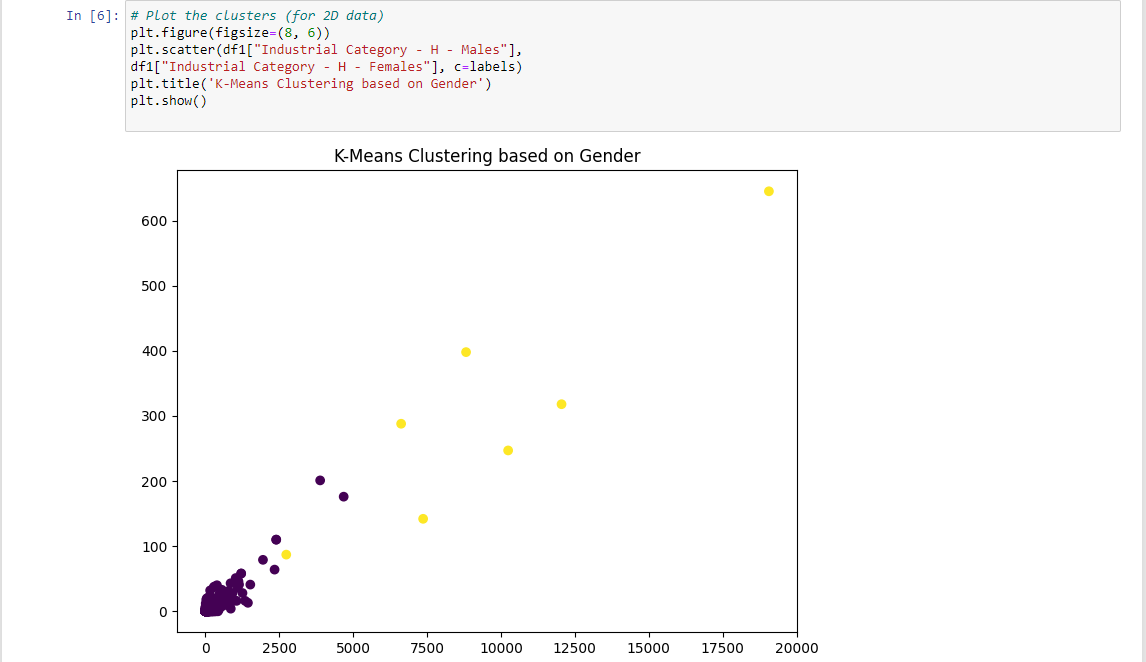
* **Data Collection**: First, you need to obtain the data you want to cluster. It is collected from <https://tn.data.gov.in/resource/marginal-workers-classified-age-industrial-category-and-sex-scheduled-caste-2011-tamil>.
* **Data Preprocessing**: This step involves cleaning and preparing the data for analysis. You may need to handle missing values, normalize the data, and convert categorical variables into numerical formats if necessary.

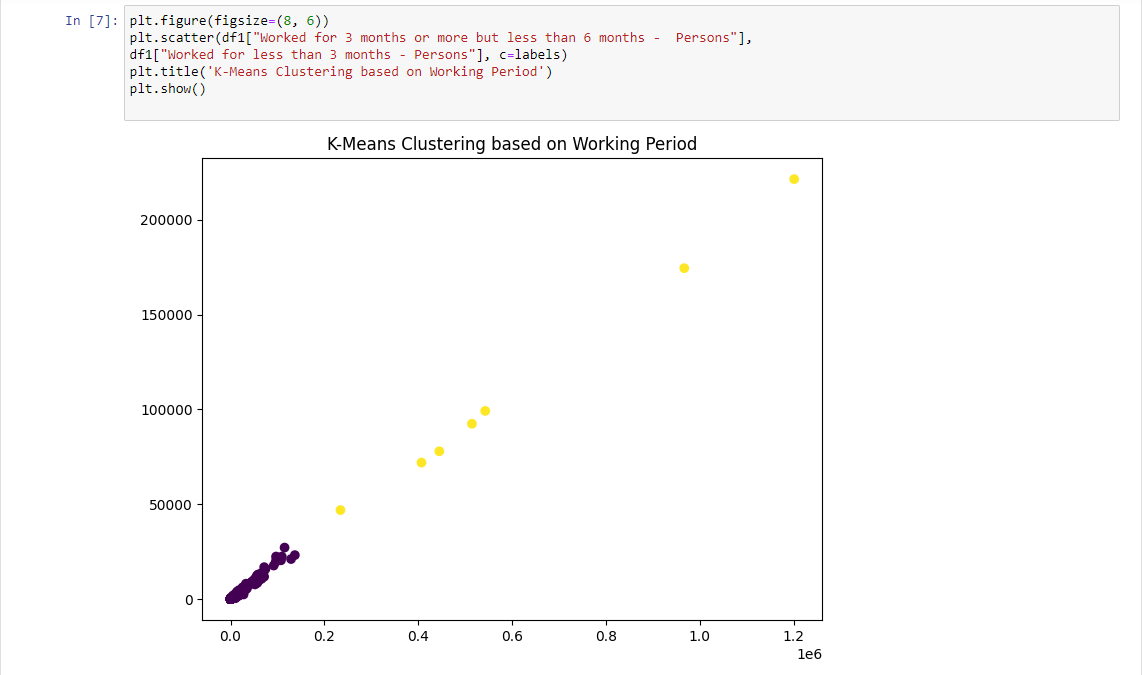


****

* **Feature Selection**: Choose the relevant features (attributes) that you want to use for clustering. Features should be selected based on the problem you're trying to solve.
* **Normalization/Standardization**: Depending on the algorithm chosen, you may need to normalize or standardize the data to ensure that features with different scales do not bias the clustering.
* **Choosing a Clustering Algorithm**: There are various clustering algorithms available, such as K-Means, Hierarchical Clustering, DBSCAN, etc. You need to select an appropriate algorithm for your dataset and problem
* **Cluster Analysis**: Apply the chosen clustering algorithm to the preprocessed data. This will group data points into clusters based on their similarity.







* **Summarizing the Model:** Summarize the model by specifying the Silhouette Score, Cluster Assignments, Cluster centers, No of clusters used, Cluster inertia etc…

**Cluster Inertia:**

Cluster inertia, also known as within-cluster sum of squares (WCSS), is a metric used to evaluate the quality of clusters in K-Means clustering or similar centroid-based clustering algorithms. It is a measure of how internally coherent the clusters are. The objective of K-Means clustering is to minimize this value

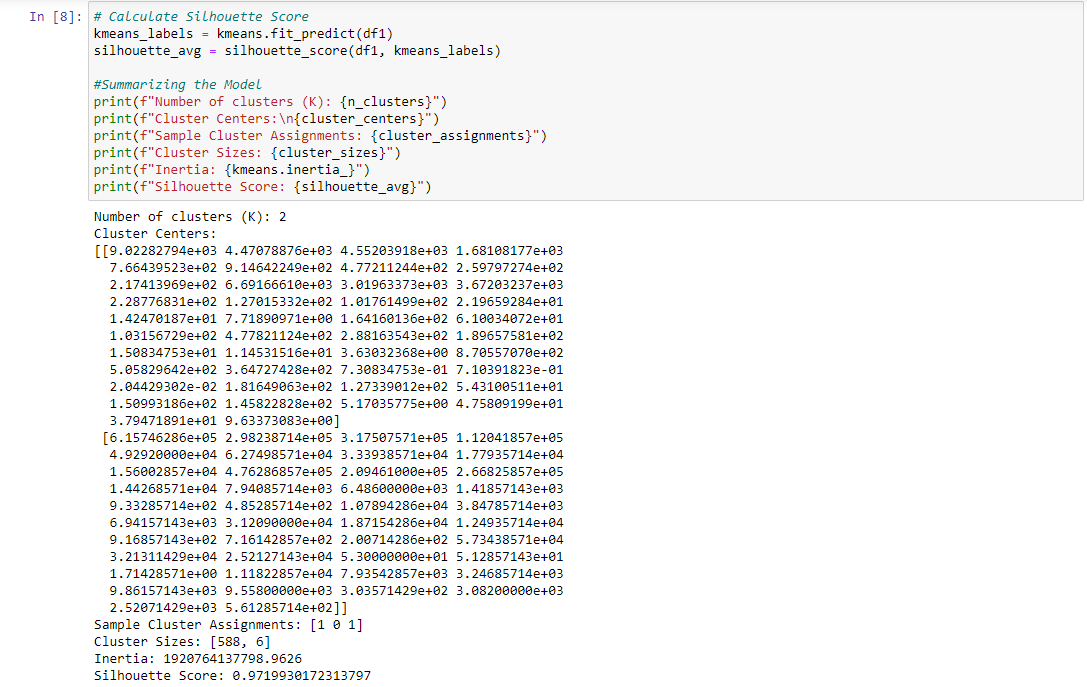
**Calculation:**

Cluster Inertia (WCSS) is calculated as the sum of the squared distances between each data point in a cluster and its centroid. The formula for calculating the WCSS for a given cluster "C" is:

WCSS(C) = Σ(dist(p, centroid(C))^2 for all p in C

Where:

* "dist(p, centroid(C))" is the Euclidean distance (or any other distance metric) between a data point "p" and the centroid of cluster "C."
* Σ denotes the summation over all data points within the cluster "C."

**Interpretation:** A lower WCSS indicates that the data points within each cluster are closer to their respective cluster centroids, meaning that the clusters are more internally cohesive. Lower WCSS values are generally desirable as they suggest better-defined and more compact clusters.

* **Insights Gained:**

**From Silhouette Score:**

The Silhouette Score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A score around 0 means that the object is on or very close to the decision boundary between two neighboring clusters and could potentially be assigned to either.

In the silhouette score calculated (0.9719930172313797), it is very close to 1, which is a high score. This indicates that the data points in your clusters are well matched to their own clusters and significantly separated from neighboring clusters, suggesting that the clustering result is excellent.

In summary, a silhouette score of approximately 0.9719930172313797 is a very good score, indicating that the clusters are well-defined and the data points within each cluster are similar to each other while being distinct from data points in other clusters.

**From Cluster Inertia:**

A WCSS value in the billions is relatively large. This suggests that the data points within the clusters are not very tightly packed around their respective centroids. In other words, the clusters may not be very internally cohesive, and the data points within each cluster are relatively spread out.

Without knowing the context and the specific number of clusters used to obtain this WCSS value, it's challenging to provide a definitive interpretation. It is required to compare this value to others, ideally on a graph, to determine whether it's relatively high or low in the context of your specific clustering problem.

**Grouping and Visualizing Data**

**Importing Libraries:**

Import necessary libraries for grouping the datasets



**Load the Dataset:**

Load the Dataset to be grouped

****

**Data Preprocessing:**

Perform data cleaning and preprocessing. This may include handling missing values, converting data types, and ensuring data quality.

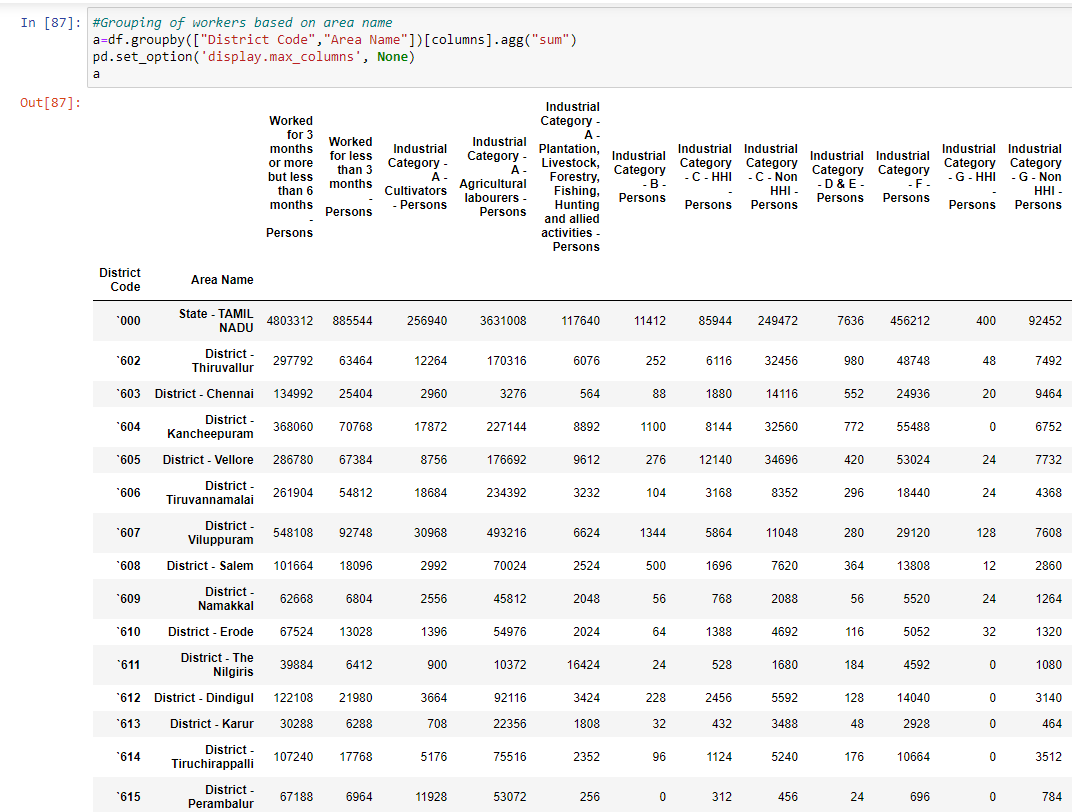
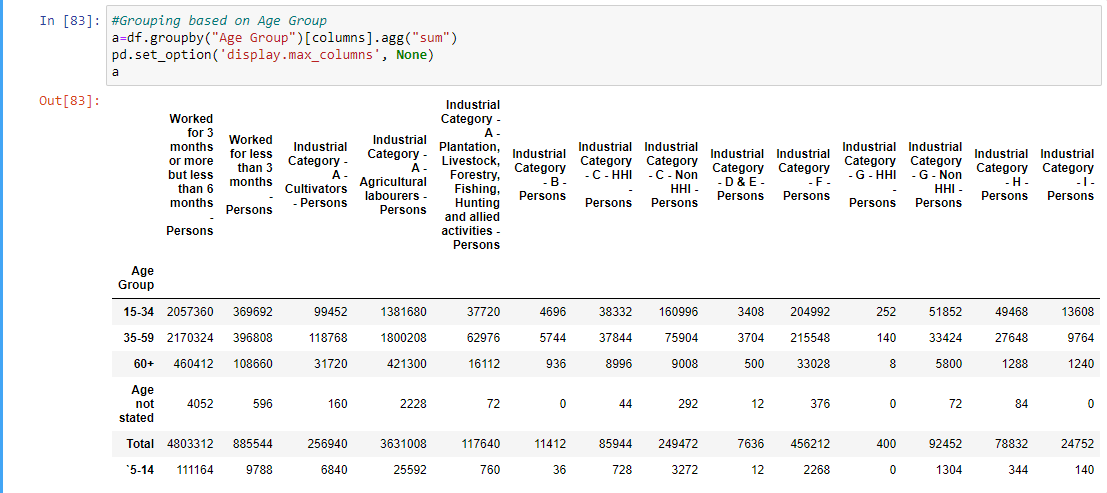
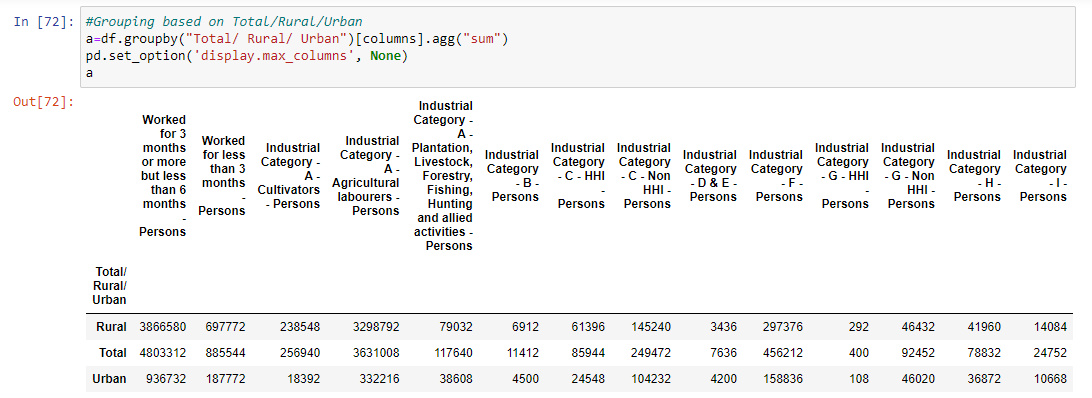
Here already the preprocessed dataset is loaded.

So no preprocessing is needed

**Grouping Data:**

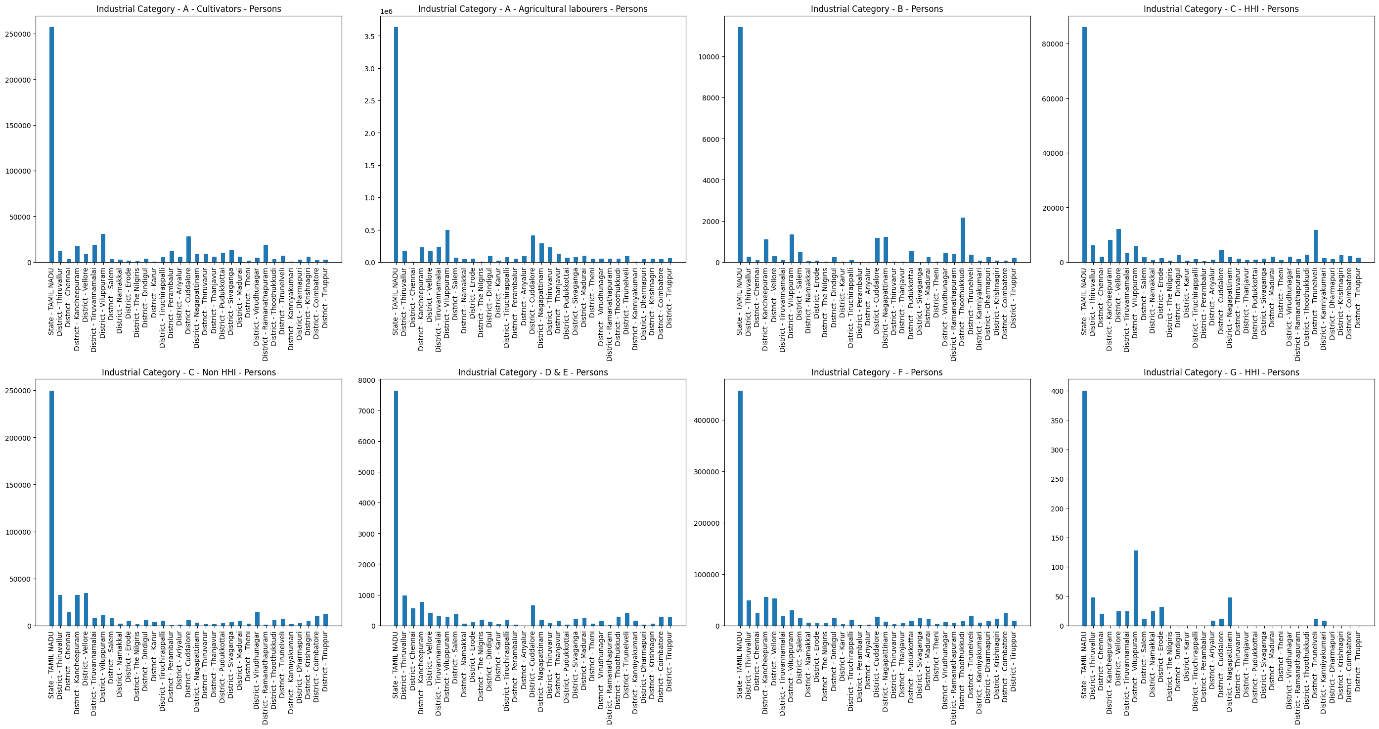
* + Identify the variables you want to group by. For example, you may want to group marginal workers by age, industrial category, or sex.
  + Use groupby operations in a data analysis library like Pandas (Python) or SQL (if your data is in a relational database) to group the data based on your chosen criteria.

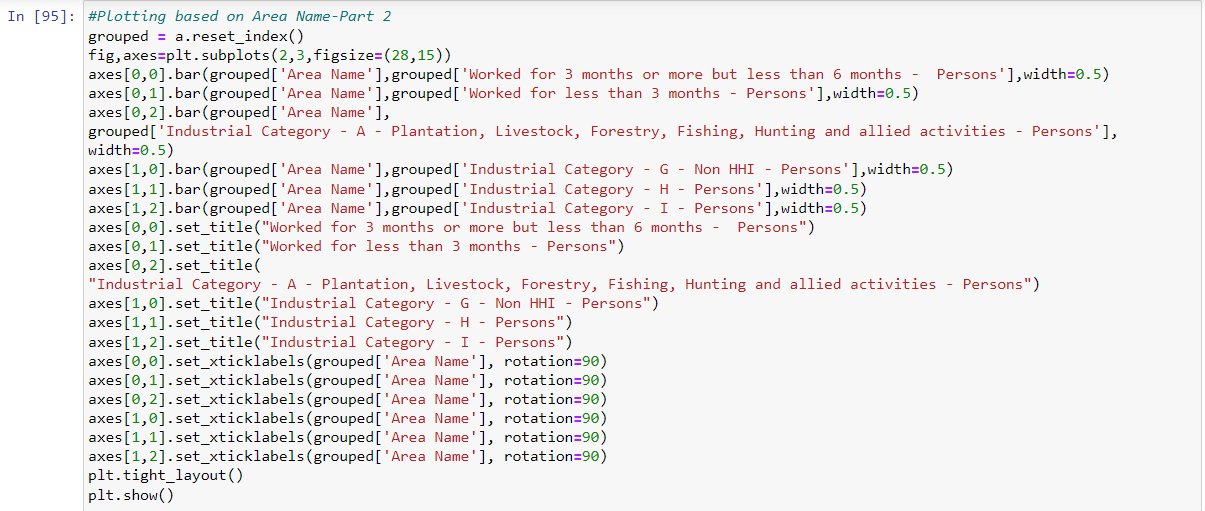


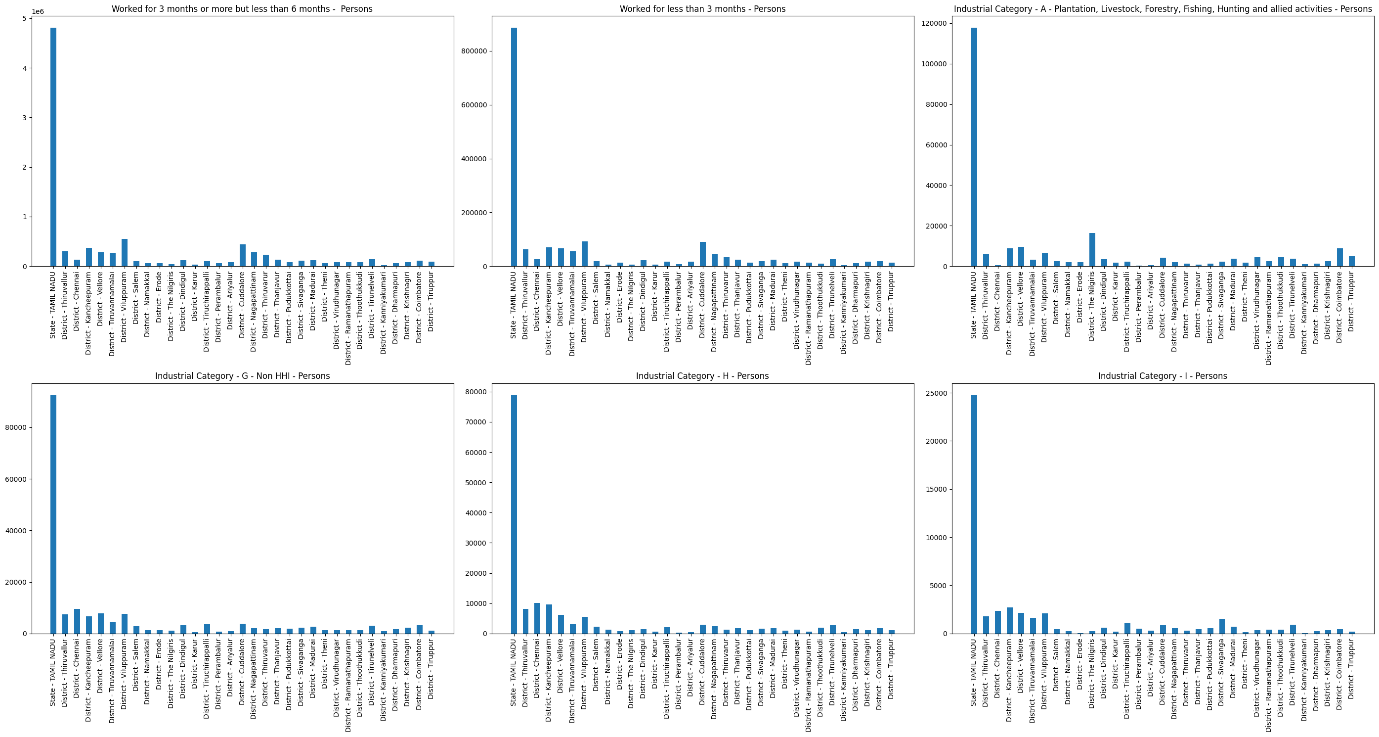
****

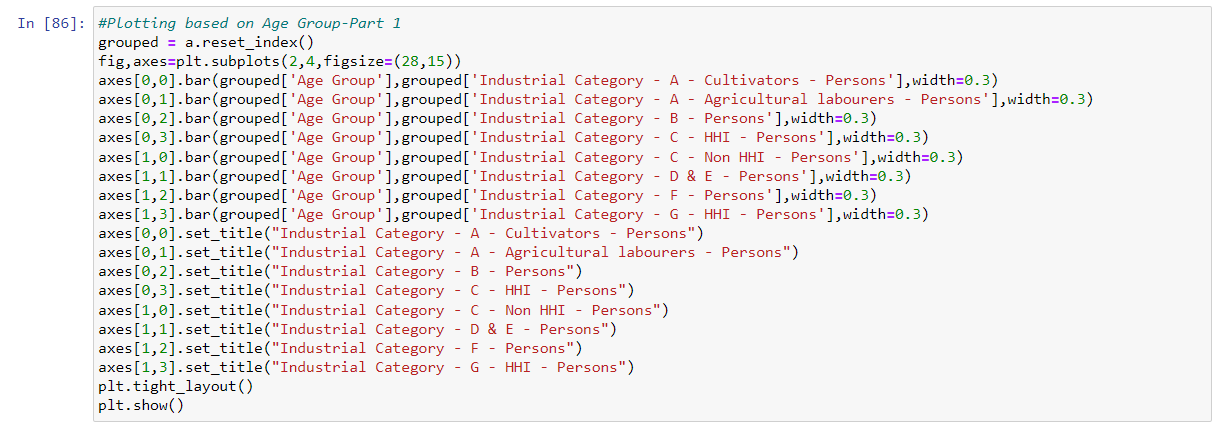
**Data Visualization:**

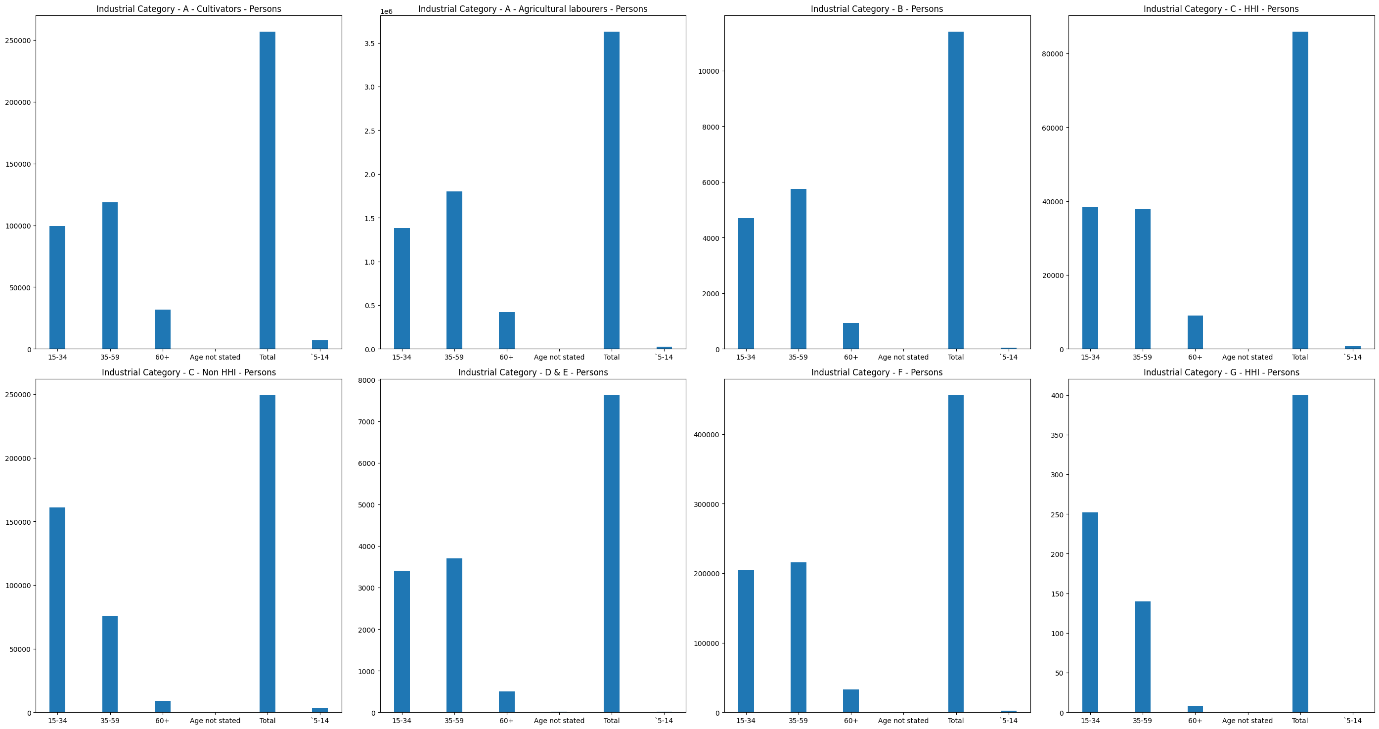
* + Create visualizations to represent the grouped data effectively. This may involve using tools like Matplotlib, Seaborn, or Plotly for Python. You can also use specialized visualization tools depending on your data type.
  + Common types of visualizations include bar charts, pie charts, histograms, and scatter plots.



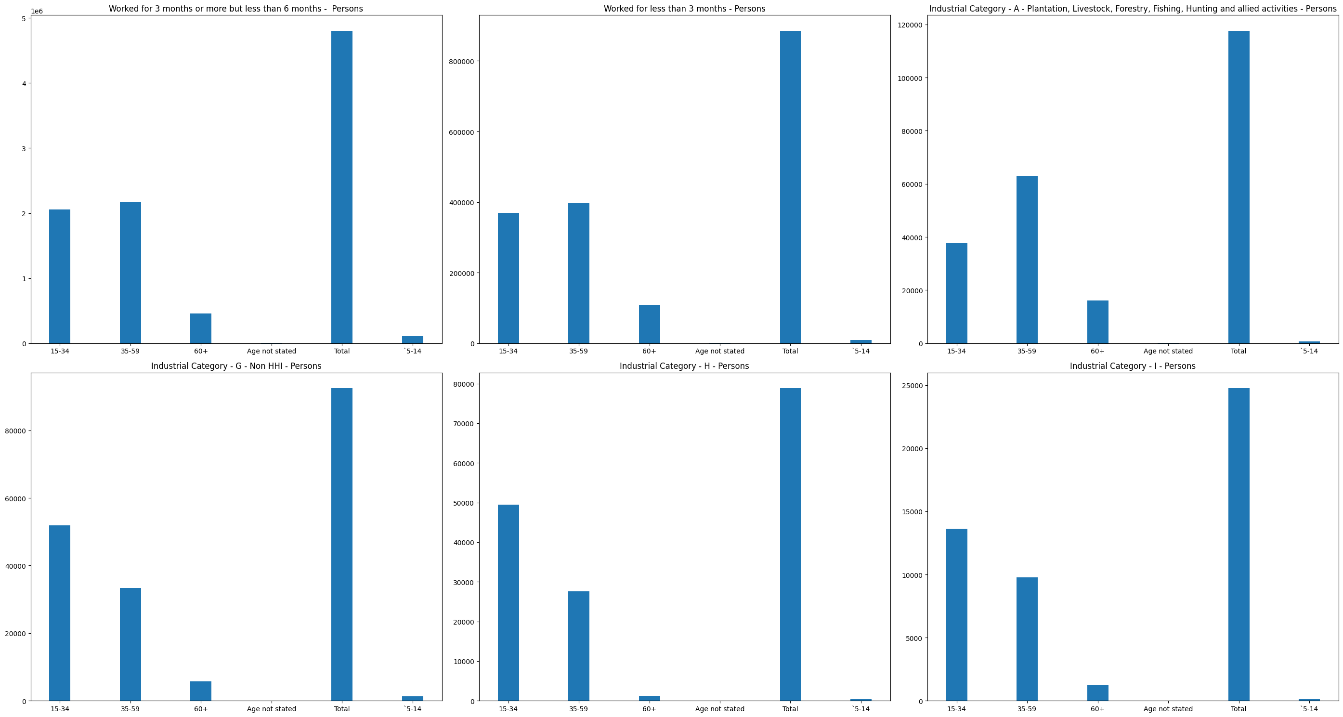


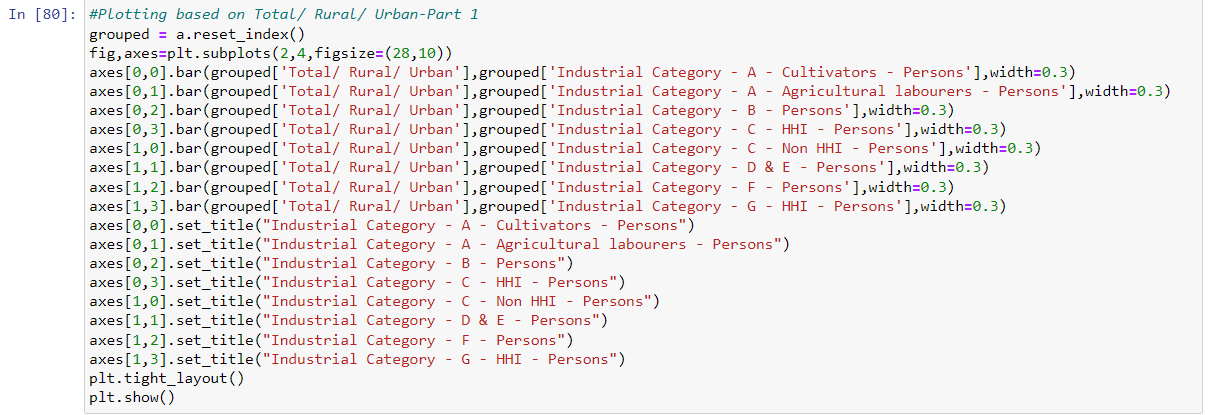
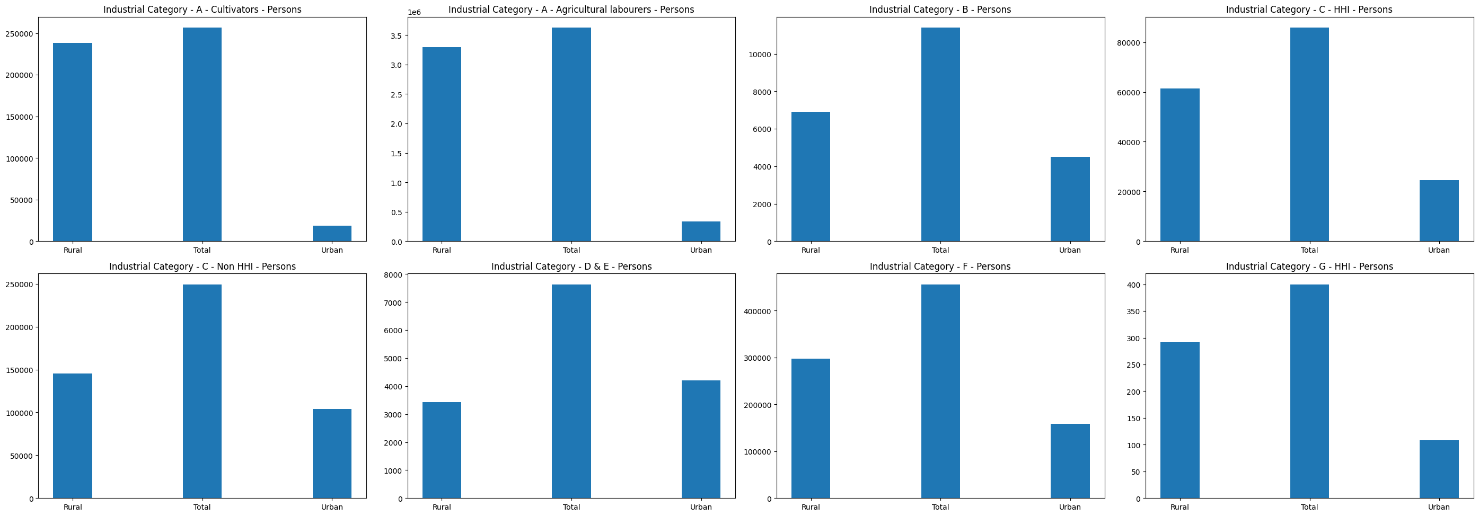


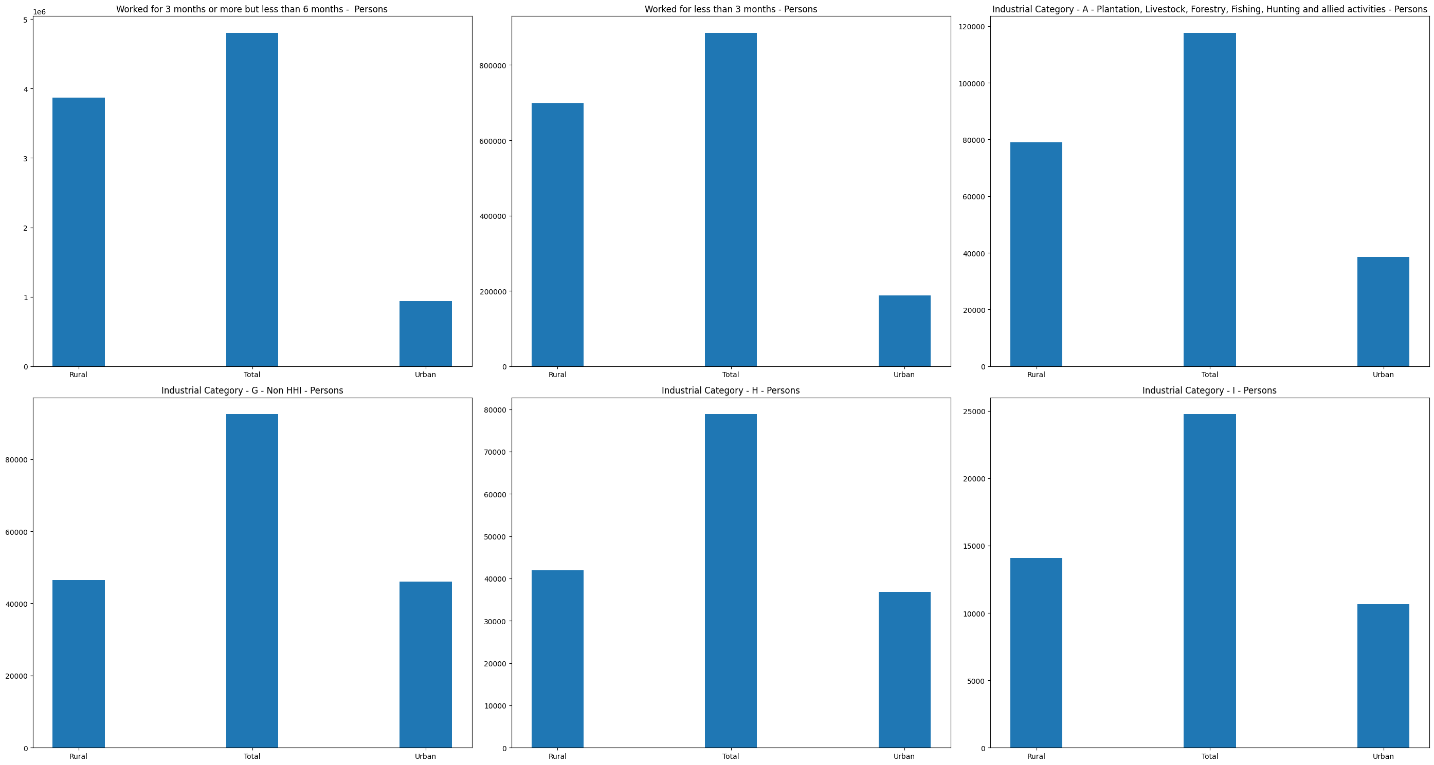








****

****

**Conclusion:**

This document outlines the process of model development and Grouping and Visualization of "Marginal Workers Classified by Age, Industrial Category, and Sex for Scheduled Caste (2011)" dataset from the Tamil Nadu government data portal