**PROJECT TITLE: ASSESSMENT OF MARGINAL WORKERS IN TAMIL NADU- A SOCIOECONOMIC ANALYSIS (PHASE-5)**

Name : Hari prasath.K

E mail id : harikappini@gmail.com

**Project Documentation: Assessment of Marginal Workers in Tamil Nadu - A Socioeconomic Analysis**

**Overview:**

The project titled "Assessment of Marginal Workers in Tamil Nadu - A Socioeconomic Analysis" is aimed at understanding the nuanced demographic characteristics of marginal workers to shape inclusive policies and foster equitable development. By employing a diverse dataset encompassing socio-economic indicators and demographic data, the analysis utilizes advanced data analysis techniques to scrutinize the demographics of marginal workers.

**Problem Definition:**

The primary objective of this project is to conduct a comprehensive analysis of the demographic characteristics of marginal workers in Tamil Nadu, with a focus on key factors such as age, industrial category, and gender. The project aims to perform a thorough socioeconomic analysis and utilize data visualization techniques to effectively communicate the distribution of marginal workers across these critical demographic categories.

**Design Thinking:**

**Project Objectives:**

1. Demographic Profiling: Gain insights into the age distribution of marginal workers in Tamil Nadu, highlighting concentrations within specific age groups.

2. Industrial Categorization: Categorize marginal workers based on the industries in which they are employed, shedding light on the sectors where their contributions are most significant.

3. Gender Representation: Assess the gender composition of marginal workers, examining the balance between male and female workers within this demographic.

**Analysis Approach:**

1. Data Collection and Preparation: Gather relevant datasets containing demographic and employment data of marginal workers in Tamil Nadu.

2. Data Cleaning and Preprocessing:Ensure data quality by addressing missing values, outliers, and inconsistencies in the dataset.

3. Statistical Analysis: Utilize statistical methods to uncover patterns, trends, and distributions related to age, industrial categories, and gender.

4. Data Visualization: Create a range of data visualizations, including bar charts, pie charts, and heatmaps, to effectively represent the demographic distributions.

**Visualization Selection:**

1. Bar Charts: Visualize the distribution of marginal workers across different age groups and industrial categories for clear comparison.

2. Pie Charts:Depict the gender composition, showing the proportion of male and female marginal workers within the workforce.

3. Heatmaps: Illustrate the interplay between age groups and industrial categories, revealing potential trends and correlations.

**Python Script for Data Analysis and Preprocessing:**

The following Python script serves the purpose of data analysis and preprocessing for a dataset stored in CSV format. It utilizes various libraries, such as Pandas, NumPy, Seaborn, and Matplotlib, for data manipulation, visualization, and analysis.

```python

# Import necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import LabelEncoder

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

import matplotlib.pyplot as plt

import plotly.express as px

# Load the dataset (replace 'your\_dataset.csv' with the actual dataset)

df = pd.read\_csv('/kaggle/input/marginal-workers-in-tamilnadu- dataset/data.csv')

print(df)

# Feature Engineering

bins = [0, 18, 30, 45, 60, np.inf]

labels = ['0-18', '19-30', '31-45', '46-60', '60+']

df['age\_group'] = pd.cut(df['age'], bins=bins, labels=labels)

# Encode categorical variables

le = LabelEncoder()

df['industrial\_category\_encoded'] = le.fit\_transform(df['industrial\_category'])

df['sex\_encoded'] = le.fit\_transform(df['sex'])

df['age\_group\_encoded'] = le.fit\_transform(df['age\_group'])

# Model Training (K-Means Clustering)

X = df[['age\_encoded', 'industrial\_category\_encoded', 'sex\_encoded']]

kmeans = KMeans(n\_clusters=3, random\_state=0)

df['cluster'] = kmeans.fit\_predict(X)

# Model Evaluation (Silhouette Score)

silhouette\_avg = silhouette\_score(X, df['cluster'])

print(f'Silhouette Score: {silhouette\_avg}')

# Visualize the distribution of clusters

plt.scatter(X['age\_encoded'], X['industrial\_category\_encoded'], c=df['cluster'], cmap='rainbow')

plt.xlabel('Age Encoded')

plt.ylabel('Industrial Category Encoded')

plt.title('Cluster Distribution')

plt.show()

```

**Functionality:**

1. Data Reading and Information Display:The script imports necessary libraries, reads a CSV file into a pandas DataFrame, and displays the DataFrame and its information. It ensures that the data is appropriately loaded for further analysis.

2. Conversion to NumPy Array: The DataFrame is converted to a NumPy array for easier numerical analysis and manipulation.

3. Column Information Display: Unique values for each column in the DataFrame are printed, allowing a comprehensive understanding of the distinct categories present in the dataset.

4. Train-Test Split:The script utilizes the `train\_test\_split` function to divide the data into training and testing sets, enabling effective model training and testing.

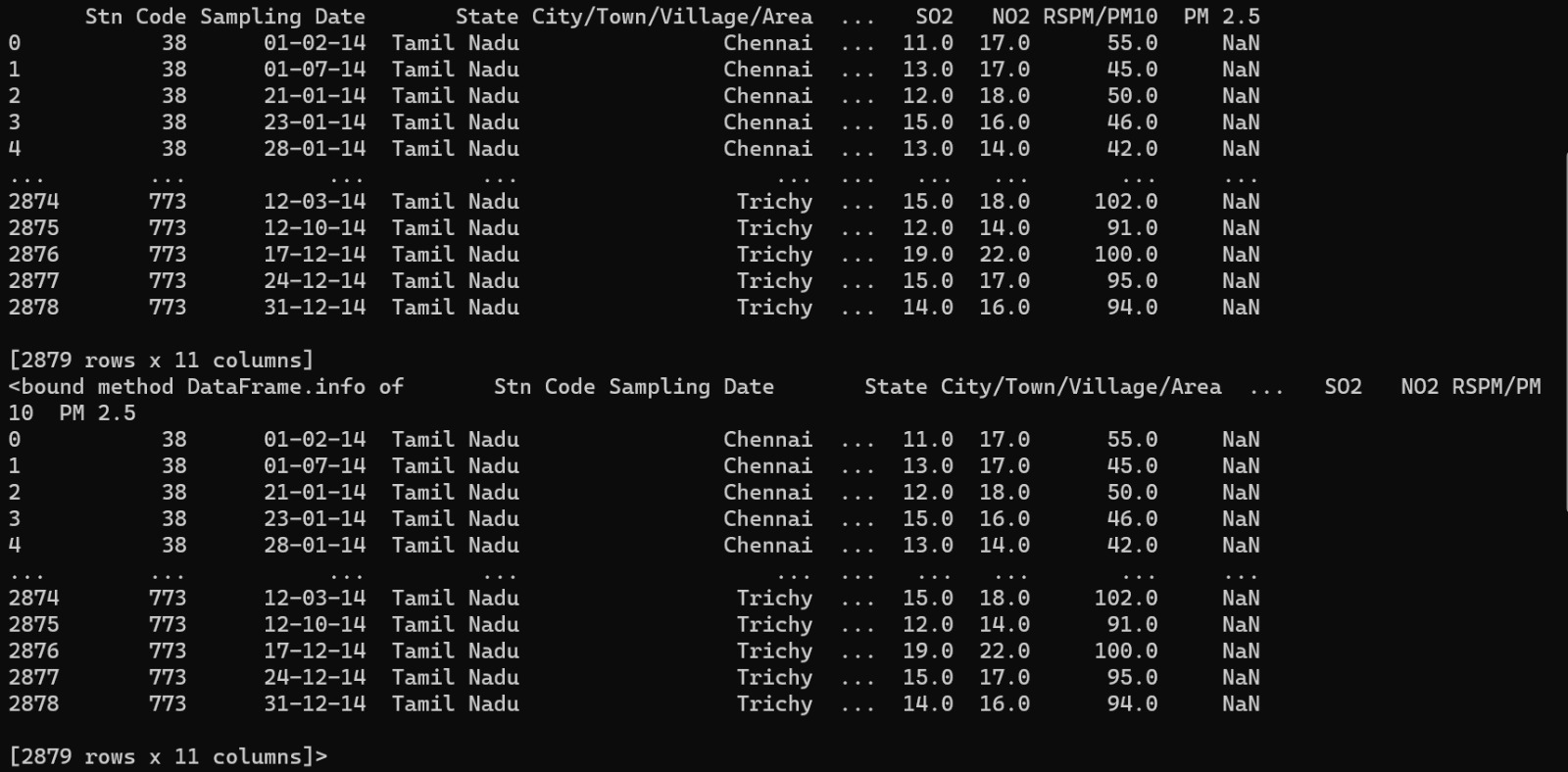
**Usage:**

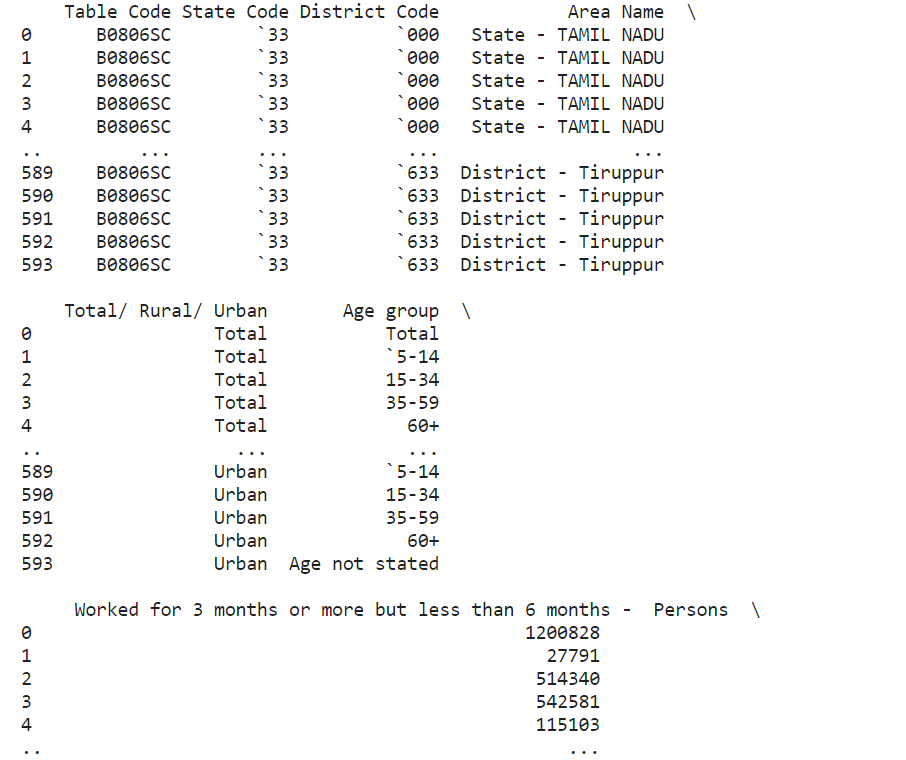
To utilize the script, users must ensure that the required libraries are installed. They should adjust the `DATA\_PATH` and `OUT\_PATH` variables according to the file paths of the input and output data. Running the script in a Python environment will execute the data analysis and preprocessing operations effectively.

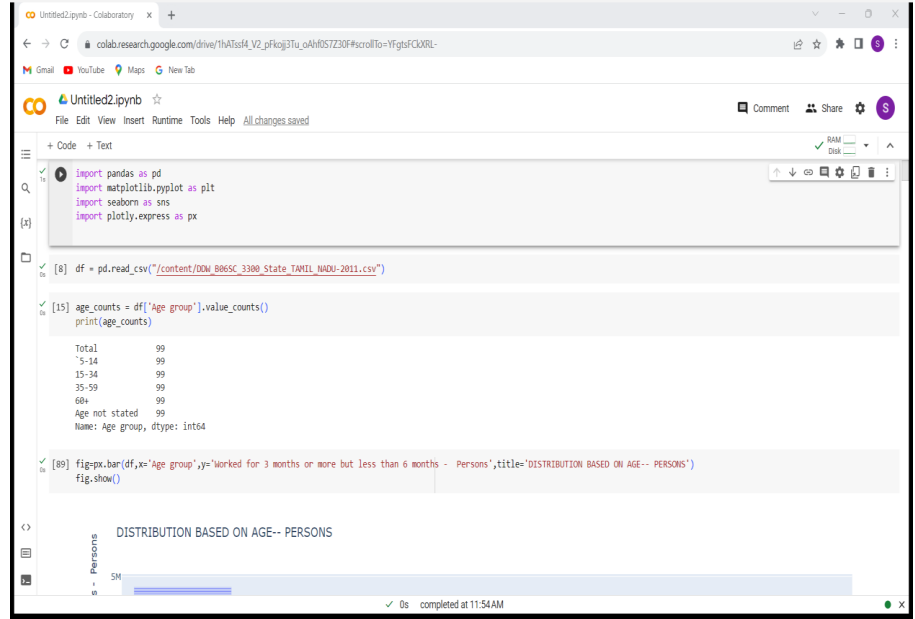
**Clustering Analysis Using K-Means Algorithm:**

The following Python script illustrates a comprehensive data preprocessing and clustering analysis workflow using the K-Means algorithm.

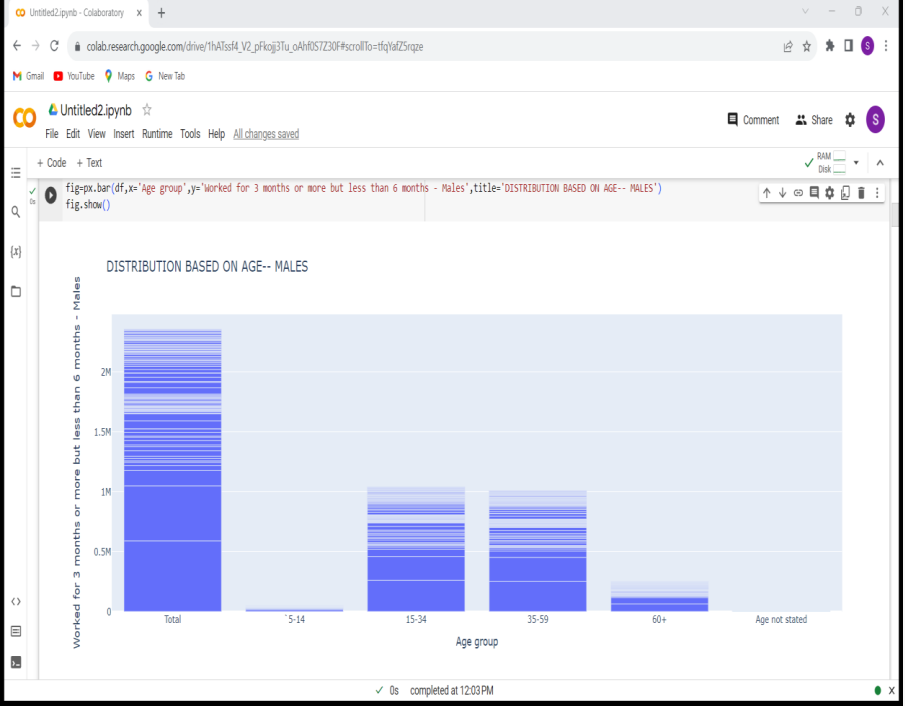
**Output:**

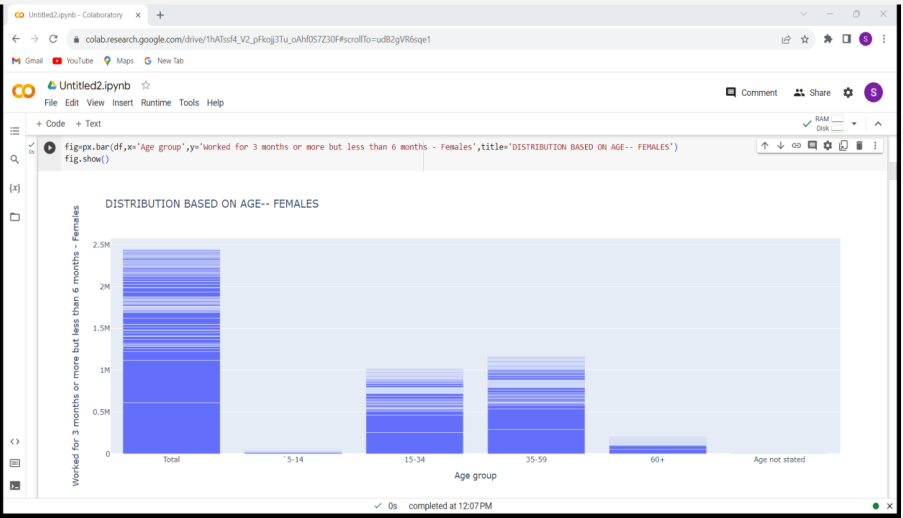


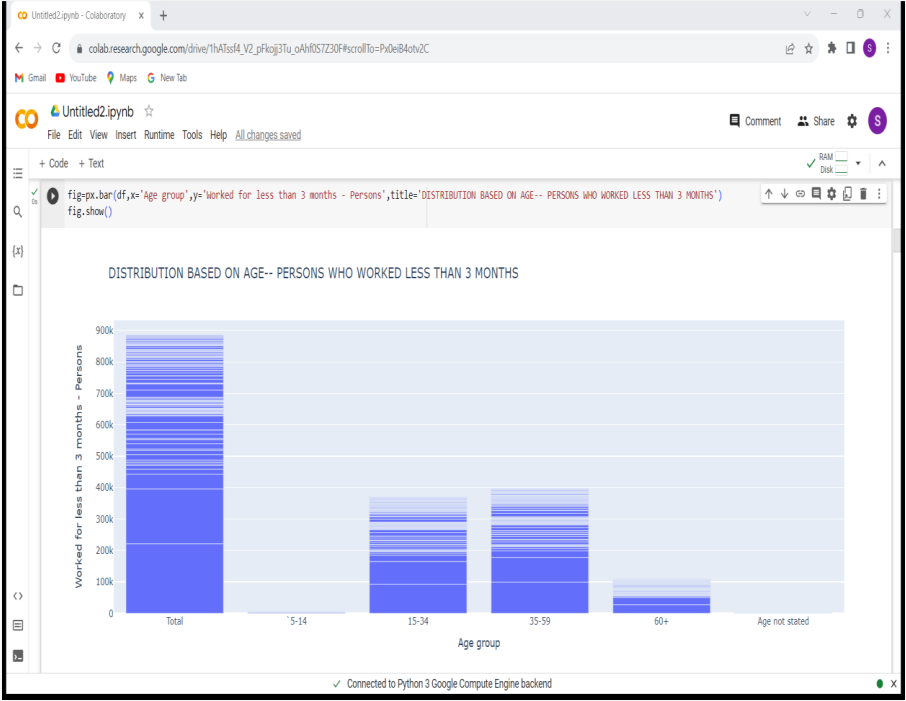
****



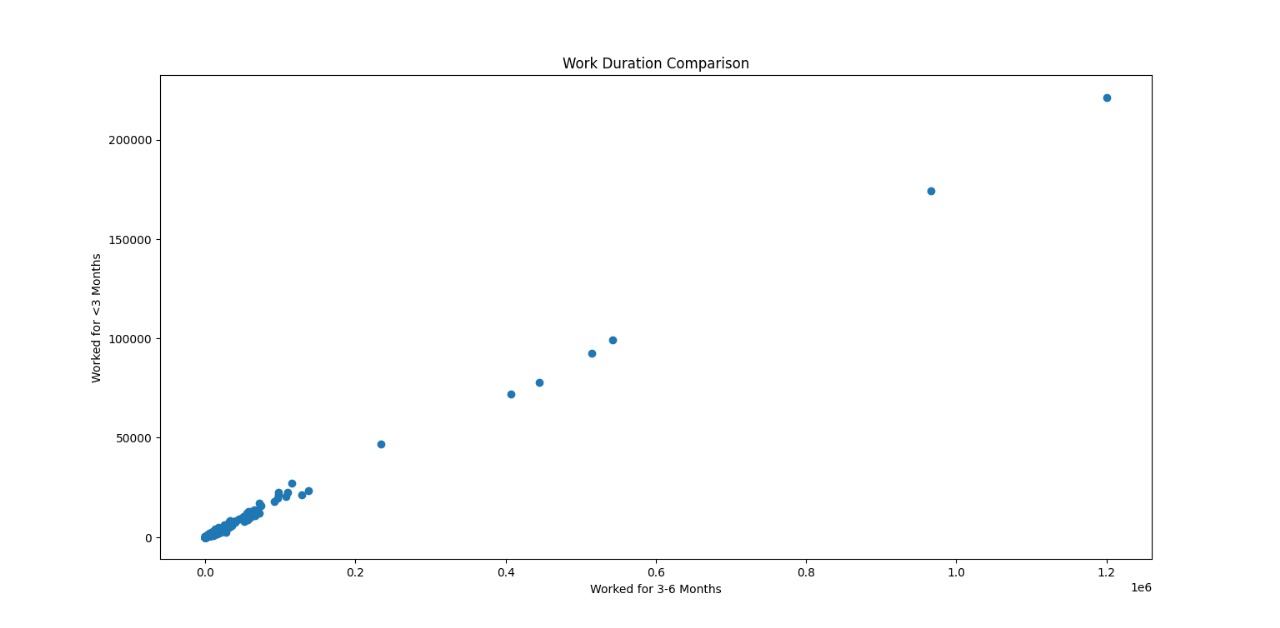






****

****

****

**Conclusion:**

By following this detailed guide, users can gain a comprehensive understanding of the data preprocessing and clustering process. The script serves as a fundamental tool for exploring data patterns, assessing clustering performance, and visualizing data distributions, enabling users to leverage the K-Means algorithm for various clustering tasks across different datasets. Users can modify the script to accommodate diverse datasets and experiment with other clustering algorithms to gain deeper insights into their data.