**PROJECT TITLE: ASSESSMENT OF MARGINAL WORKERS IN TAMIL NADU- A SOCIOECONOMIC ANALYSIS (PHASE-4)**

**Name** :Hari Prasath.K

**E mail id** :harikappini@gmail.com

**Overview**

This Python script is designed to illustrate a comprehensive data preprocessing and clustering analysis workflow using the K-Means algorithm. The script provides a step-by-step guide on how to handle data, perform necessary transformations, and evaluate the clustering results. It aims to facilitate a better understanding of the data distribution and the performance of the clustering algorithm.

**Importing Necessary Libraries**

The script begins by importing essential libraries required for data manipulation, analysis, visualization, and machine learning. These include `pandas` for data handling, `numpy` for numerical operations, `scikit-learn` for machine learning algorithms, and `matplotlib` for data visualization.

**Loading the Dataset**

The script loads the dataset from a CSV file using the `pd.read\_csv()` function and stores it in a pandas DataFrame for further analysis. Users need to replace the placeholder 'your\_dataset.csv' with the actual dataset file name.

**Feature Engineering**

To enrich the dataset, the script demonstrates an example of feature engineering by creating distinct age groups based on predefined bins. The `pd.cut()` function is utilized to segment the 'age' column into specific age groups, which are then added as a new categorical feature, 'age\_group', in the DataFrame.

**Encoding Categorical Variables**

Categorical variables in the dataset are encoded using the `LabelEncoder` from scikit-learn. The script encodes the 'industrial\_category' and 'sex' columns, creating new columns, 'industrial\_category\_encoded' and 'sex\_encoded', respectively, to represent the encoded values.

**Model Training (K-Means Clustering)**

For the purpose of clustering analysis, the script prepares the data by selecting specific features from the DataFrame and employing the K-Means clustering algorithm. The selected features are used to train the K-Means model, which assigns each data point to a particular cluster based on its characteristics.

**Model Evaluation (Silhouette Score)**

To assess the clustering quality, the script calculates the Silhouette Score, a metric that quantifies the cohesion and separation of clusters. The script prints the Silhouette Score to provide insights into the effectiveness of the clustering algorithm in capturing inherent patterns within the data.

**Visualization of Clusters**

The script generates a scatter plot to visualize the distribution of data points within the clusters. The plot represents the relationship between the encoded age and industrial category, with each data point colored according to the cluster it belongs to. This visualization aids in the interpretation of the clustering results and the underlying data structure.

**Code**

**# 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')**

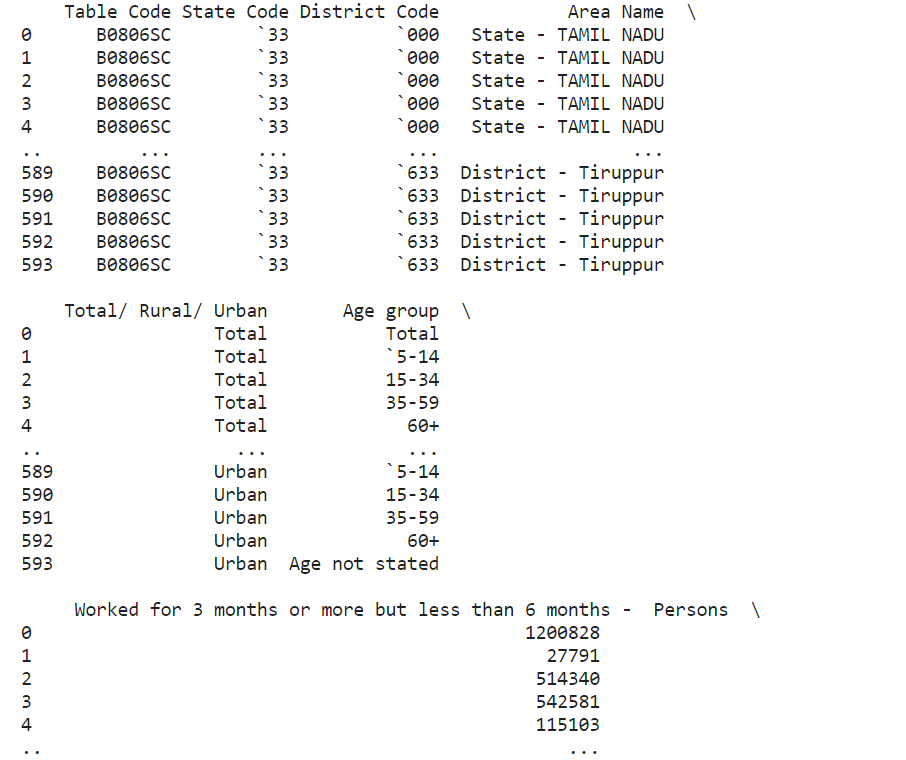
**plt.xlabel('Age Encoded')**

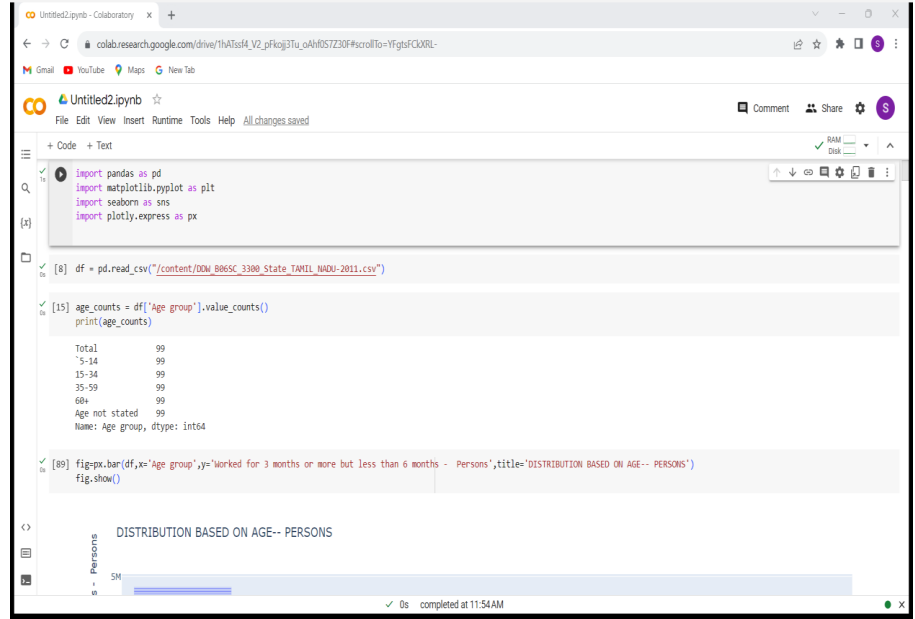
**plt.ylabel('Industrial Category Encoded')**

**plt.title('Cluster Distribution')**

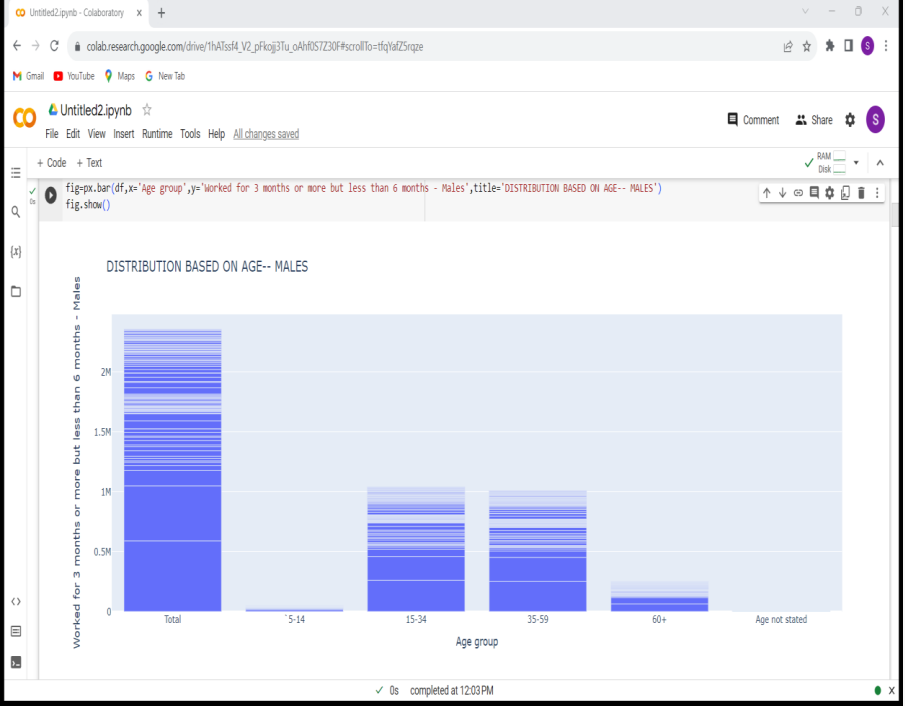
**plt.show()**

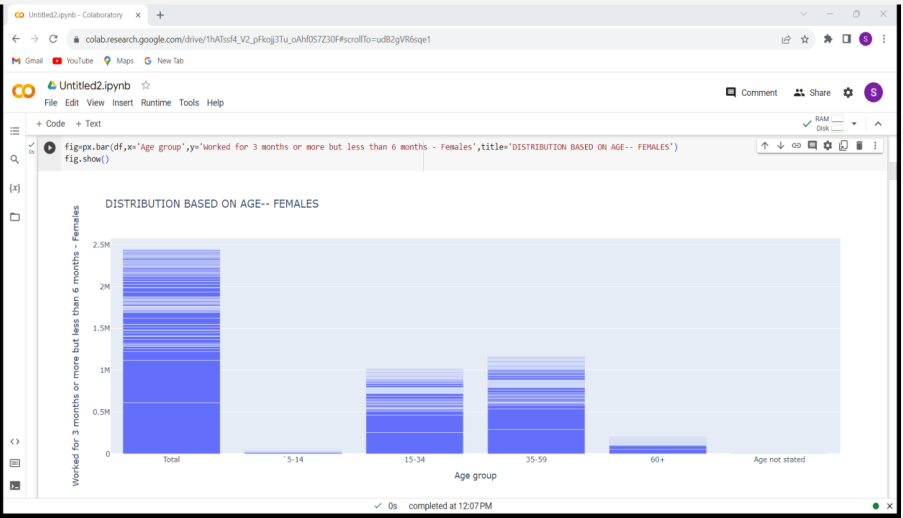
**Output**

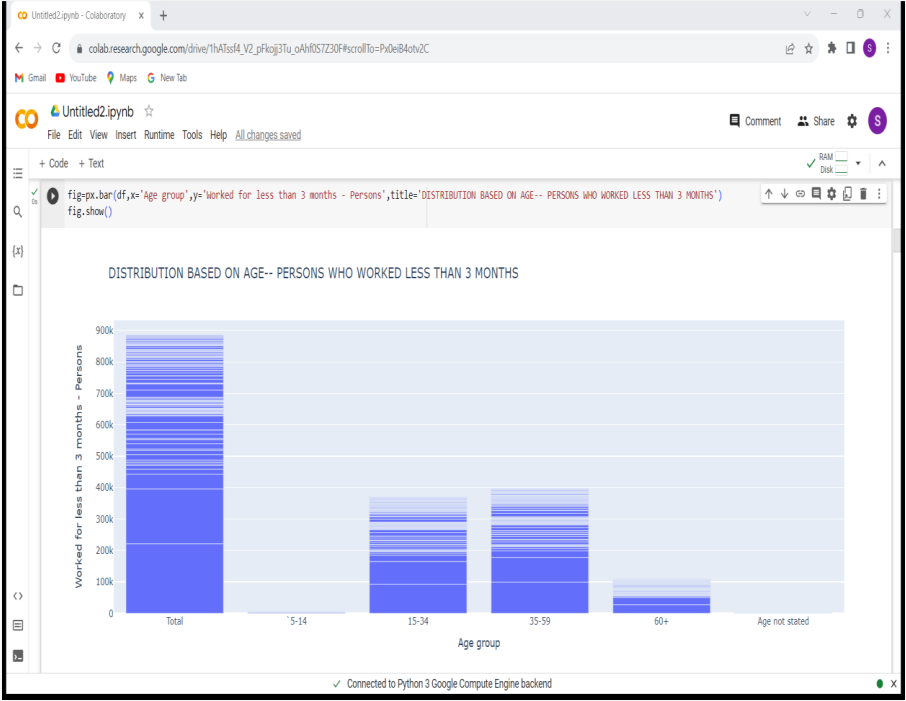
****









****

****

**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.