## TN Marginal Workers Assessment PHASE 3:DEVELOPMENT PART 1

#### Introduction:

Data loading and pre-processing are fundamental steps in any data-driven project, whether it's machine learning, data analysis, or any other data-centric task. These crucial processes involve gathering raw data from various sources and transforming it into a suitable format for analysis or model training

Proper data preprocessing is essential to ensure that the data used for analysis or machine learning is accurate, consistent, and suitable for the intended purpose

Data preprocessing is the process of cleaning and preparing raw data before it is used in data analysis.

### The steps are:

- Data Cleaning
- Data Transformation
- Data Reduction
- Data Splitting

### Data Cleaning:

Removing or correcting any errors or inconsistencies in the data, such as missing values, duplicates, or outliers.

## Data Transformation:

This can include converting data into a suitable format, scaling or normalizing features, and encoding categorical variables into numerical values.

#### Data Reduction:

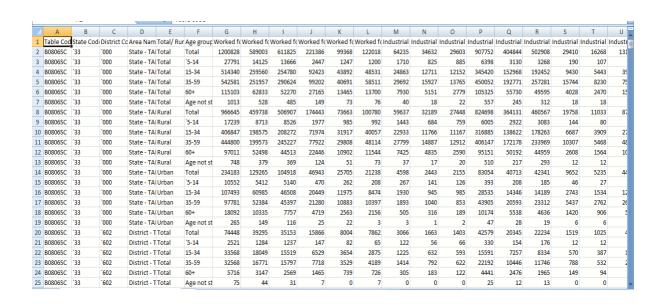
Sometimes, data may be too large or contain redundant information. Data reduction techniques, like dimensionality reduction, can be applied to reduce the data's complexity.

## Data Splitting:

Dividing the data into training and testing sets for model evaluation and validation.

The given data set:

Dataset Link: <a href="https://tn.data.gov.in/resource/marginal-workers-classified-age-industrial-category-and-sex-scheduled-caste-2011-tamil">https://tn.data.gov.in/resource/marginal-workers-classified-age-industrial-category-and-sex-scheduled-caste-2011-tamil</a>



## Algorithm to perform data loading and cleaning

Step1: Import necessary libraries:

import pandas as pd

Step2: Load the data:

Use pd.read\_csv()

Step3: Explore the data:

Use functions like head(), info(), and describe()

Step4: Data Cleaning:

Identify and address missing values, duplicates, and other data quality issues.

Step5: Data Transformation

Step6: Save the cleaned data

#### PROGRAM:

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

# Import the data set

df =

 $pd.read\_csv("C:\\Users\lenovo\Downloads\DDW\_B06SC\_330\ O\_State\_TAMIL\_NADU-2011.csv")$ 

# Drop columns that aren't useful

df.drop('Sl. No.', axis=1, inplace=True)

```
# Handle missing data
# In this case, we will drop the rows with missing values.
df.dropna(inplace=True)
# Encode categorical data
# The data set contains categorical data in the `Industrial
category' column. We can encode this data using the
`pd.get_dummies()` function.
df = pd.get_dummies(df, columns=['Industrial category'])
# Split the data set into training and test sets
# We will split the data set into 80% training data and 20%
test data.
X_train, X_test, y_train, y_test =
train_test_split(df.drop('Scheduled Caste', axis=1),
df['Scheduled Caste'], test_size=0.2, random_state=42)
# Feature scaling
# The data set contains features with different scales, so we
will scale them using the 'StandardScaler()' class.
scaler = StandardScaler()
scaler.fit(X_train)
X_{train} = scaler.transform(X_{train})
X_{test} = scaler.transform(X_{test})
# Save the preprocessed data set
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)
```

To start the data analysis by loading and pre-processing the dataset.

#### ALGORITHM:

## Step 1: Import libraries

import pandas as pd
import matplotlib.pyplot as plt

#### Step 2: Load the dataset.

```
df = \\ pd.read\_csv("C:\\\begin{subarray}{c} DOW\_B06SC\_33 \\ OO\_State\_TAMIL\_NADU-2011.csv") \\ \end{array}
```

# **Step 3:** Create a bar chart of the number of marginal workers by age group

```
plt.figure(figsize=(10, 6))

df['Age'].value_counts().plot(kind='bar',
color=plt.cm.viridis.colors)

plt.xlabel('Age group')

plt.ylabel('Number of marginal workers')

plt.title('Number of marginal workers by age group')
```

plt.show()

**Step 4:** Create a pie chart of the number of marginal workers by industrial category

```
plt.figure(figsize=(10, 6))

df['Industrial category'].value_counts().plot(kind='pie',
autopct="%1.1f%%")

plt.title('Percentage of marginal workers by industrial
category')

plt.show()
```

**Step 5:** Create a stacked bar chart of the number of marginal workers by sex and industrial category

#### PROGRAM:

```
import pandas as pd
import matplotlib.pyplot as plt
df =
pd.read_csv("C:\\Users\lenovo\Downloads\DDW_B06SC_330
0_State_TAMIL_NADU-2011.csv")
plt.figure(figsize=(10, 6))
df['Age'].value_counts().plot(kind='bar',
color=plt.cm.viridis.colors)
plt.xlabel('Age group')
plt.ylabel('Number of marginal workers')
plt.title('Number of marginal workers by age group')
plt.show()
plt.figure(figsize=(10, 6))
df['Industrial category'].value_counts().plot(kind='pie',
autopct="%1.1f%%")
plt.title('Percentage of marginal workers by industrial
category')
plt.show()
plt.figure(figsize=(10, 6))
```

```
df.groupby(['Industrial category',
'Sex'])['Age'].count().unstack().plot(kind='bar',
stacked=True)
plt.xlabel('Industrial category')
plt.ylabel('Number of marginal workers')plt.title('Number of
marginal workers by sex and industrial category')
plt.legend()
plt.show()
```