ADS PHASE2

ASSESSMENT OF MARGINAL WORKERS IN TAMILNADU- A SOCIOECONOMIC ANALYSIS

Details about the project working flow:

1.Explanation about the marginal workers:

A marginal worker is typically defined as a person who is engaged in irregular or occasional employment, or who works for a limited number of hours per day or days per week. This term is often used in the context of labour statistics and economic analysis to distinguish individuals who have less stable or consistent employment compared to regular or full-time workers. Marginal workers may include part-time employees, casual labourers, seasonal workers, or those who work intermittently. The concept of marginal workers is important for understanding labour market dynamics and socioeconomic conditions.

2.Details about the dataset:

In our domain (APPLIED DATA SCIENCE), we are using the dataset which has been given by skill up team.

(i.e.,) **Dataset Link:** https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states

3.Details about the column used:

For this project, we are using certain columns such as Age group,District code,State code, Area Name, Worked for 3 months or more but less than 6 months - Persons, Worked for 3 months or more but less than 6 months – Males, Worked for 3 months or more but less than 6 months – Females, Worked for less than 3 months – Persons, Worked for less than 3 months – Males, Worked for less than 3 months – Females, Industrial Category - A - Cultivators – Persons, Industrial Category - A - Cultivators – Males, Industrial Category - A - Cultivators – Females, Industrial Category - A - Agricultural labourers – Persons, Industrial Category - A - Agricultural labourers – Males, Industrial Category - A - Agricultural labourers – Females, Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities – Persons, Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities – Males, Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities – Females,etc.,

4.Details of libraries to be used and way to download:

The libraries used in a dataset typically refer to the software libraries or packages that were used to create, manipulate, analyze, or visualize the data. The specific libraries used can vary widely depending on the tools and programming languages employed in the data collection and analysis process.

Common libraries for data handling and analysis include:

Python Libraries:

Pandas: For data manipulation and analysis.

NumPy: For numerical operations.

Matplotlib and Seaborn: For data visualization.

Scikit-learn: For machine learning.

TensorFlow and PyTorch: For deep learning.

To read, clean, train, test, evaluate accuracy, and visualize datasets, you can use a combination of various libraries and tools, depending on your programming language of choice (commonly Python) and the specific requirements of your project. Here's a list of libraries often used for each of these tasks:

Reading Data:

Pandas: It's the most common library for reading and manipulating data in various formats, such as CSV, Excel, SQL databases, and more.

Data Cleaning:

Pandas: It's also excellent for data cleaning, including handling missing values, removing duplicates, and transforming data.

Training and Testing:

Scikit-learn (sklearn): This library provides a wide range of machine learning algorithms for training models. It also includes tools for data splitting and cross-validation.

TensorFlow and PyTorch: If you're working with deep learning, these libraries are widely used for training and testing neural networks.

Evaluating Accuracy:

For classification tasks, you can use libraries like Scikit-learn for various metrics such as accuracy, precision, recall, F1-score, etc.

For regression tasks, metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared can be calculated using Scikit-learn.

Data Visualization:

Matplotlib and Seaborn: These are widely used for creating static and interactive visualizations in Python.

Plotly and Bokeh: If you need interactive visualizations, these libraries can be very useful.

Here's a basic example of how you might use these libraries in Python:

```
import pandas as pd

from sklearn.model_selection import train_test_split
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
        # Reading data
data = pd.read_csv('your_dataset.csv')
        # Data cleaning (e.g., handling missing values)
data.dropna(inplace=True)
        # Splitting data into training and testing sets
X = data.drop('target', axis=1)
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
        # Training a model
model = RandomForestClassifier()
model.fit(X_train, y_train)
        # Testing the model
y_pred = model.predict(X_test)
        # Calculating accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
        # Data visualization
plt.scatter(data['feature1'], data['feature2'])
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

This is a simplified example, and in practice, the code can be more complex depending on the dataset and machine learning tasks involved.

5.Way to download libraries in python:

In Python, you can download and install libraries using a package manager called pip. Here's how you can use pip to download and install libraries:

Installing a Library:

To install a library, open your command prompt or terminal and run the following command:

pip install library_name

6. How to train and test the dataset:

```
        # Splitting data into training and testing sets
X = data.drop('target', axis=1)
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
        # Training a model
model = RandomForestClassifier()
model.fit(X_train, y_train)
        # Testing the model
y_pred = model.predict(X_test)
```

7.What metrics used for check the accuracy in dataset:

To calculate accuracy, you can use the accuracy_score function from Scikit-learn. It measures the proportion of correctly predicted instances out of the total instances in the dataset.

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_true, y_pred)
```

8.Steps involved in this project:

1.Import libraries

2.Load data (i.e., ‘pd.read_csv()’)

3.Preprocessing the data (i.e., to handling missing data,drop the unused column,rows etc.,)

4.Split the data (i.e., train_test_split())

5.Train the model(i.e., ‘X_train’ and ‘Y_train’)

6.Test the model(i.e., X_test)

7.Evaluate model performance (i.e., accuracy,precision,etc.,)