DATA ANALYTICS WITH COGNOS

**PHASE 3:** DEVELOPMENT PART 1

***PROJECT TITLE: MARGINAL WORKERS ASSESSMENT IN TN SOCIO ECONOMIC ANALYSIS***

**MARGINAL WORKERS ASSESSMENT**

**Introduction:**

* Marginal workers are those whose employment is insecure, poorly paid, and unprotected. They often work in informal sectors such as construction, agriculture, and domestic work. Marginal workers are vulnerable to exploitation and abuse, and they often lack access to social security benefits.
* Marginal workers refer to individuals who are employed in low-productivity or low-income jobs and are often at risk of underemployment or unemployment.
* Assessing marginal workers involves evaluating their economic status, job stability, and access to essential benefits, with the aim of understanding their employment conditions and identifying ways to improve their livelihoods.
* This assessment can include factors such as income levels, job security, education, and access to social services, and is often used to inform labor market and social policy decisions.

**GIVEN DATASET :**

Table Code State Code District Code Area Name Total/ Rural/ Urban \

0 B0806SC `33 `000 State - TAMIL NADU Total

1 B0806SC `33 `000 State - TAMIL NADU Total

2 B0806SC `33 `000 State - TAMIL NADU Total

3 B0806SC `33 `000 State - TAMIL NADU Total

4 B0806SC `33 `000 State - TAMIL NADU Total

Age group Worked for 3 months or more but less than 6 months - Persons \

0 Total 1200828

1 `5-14 27791

2 15-34 514340

3 35-59 542581

4 60+ 115103

Worked for 3 months or more but less than 6 months - Males \

0 589003

1 14125

2 259560

3 251957

4 62833

Worked for 3 months or more but less than 6 months - Females \

0 611825

1 13666

2 254780

3 290624

4 52270

Worked for less than 3 months - Persons ... \

0 221386 ...

1 2447 ...

2 92423 ...

3 99202 ...

4 27165 ...

Industrial Category - R to U - Non HHI - Males \

0 55801

1 9774

2 32803

3 9675

4 3394

3671

[5 rows x 69 columns]



​

**Necessary step to follow:**

**1.Import Libraries:**

Start by importing the necessary libraries:

**Program:**

Import pandas as pd

Import numpy as np

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

from sklearn. preprocessing import StandardScaler

**2.Load the dataset:**

Load your dataset into a Pandas DataFrame .You can typically find marginal workers datasets in CSV format ,but you can adapt this code to other formats as needed.

**Program:**

df=pd.read\_csv('E:\USA\_Housing.csv')

Pd.read()

**3.ExploratoryDataAnalysis(EDA):**

Perform EDA to understand your data better.This includes checking for missing values, exploring the data's statistics ,and visualizing it to identify patterns.

**Program:**

#Checkformissingvalues

print(df.isnull().sum())

#Explorestatistics

print(df.describe())

#Visualizethedata(e.g.,histograms,scatterplots,etc.)

**4.Feature Engineering:**

Depending on your dataset ,you may need to create new features or transform existing ones. This can involve one-hot encoding categorical Variables ,handling date/time data ,or scaling numerical features.

**Program:**

#Example:One-hotencodingforcategoricalvariables

df=pd.get\_dummies(df,columns=['Avg.AreaIncome','Avg.Area

HouseAge'])

**5.SplittheData**:

Split your data set into training and testing sets. This helps you evaluate your model's performance later.

X=df.drop('price',axis=1)#Features

y=df['price']#Targetvariable

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,

random\_state=42)

**6. Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 andstd=1) is a common choice.

Program:

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Output:**

Table Code 0

State Code 0

District Code 0

Area Name 0

Total/ Rural/ Urban 0

..

Industrial Category - R to U - HHI - Males 0

Industrial Category - R to U - HHI - Females 0

Industrial Category - R to U - Non HHI - Persons 0

Industrial Category - R to U - Non HHI - Males 0

Industrial Category - R to U - Non HHI - Females 0

Length: 69, dtype: int64

Worked for 3 months or more but less than 6 months - Persons \

count 5.940000e+02

mean 1.617277e+04

std 7.607172e+04

min 0.000000e+00

25% 2.872500e+02

50% 2.225500e+03

75% 9.628500e+03

max 1.200828e+06

Worked for 3 months or more but less than 6 months - Males \

count 594.000000

mean 7932.700337

std 36864.822704

min 0.000000

25% 147.250000

50% 1147.000000

75% 4770.500000

max 589003.000000

Worked for 3 months or more but less than 6 months - Females \

count 594.000000

mean 8240.067340

std 39259.545337

min 0.000000

25% 144.000000

50% 1076.000000

75% 4887.500000

max 611825.000000

Worked for less than 3 months - Persons \

count 594.000000

mean 2981.629630

std 13909.621137

min 0.000000

25% 27.000000

50% 430.000000

75% 1775.250000

max 221386.000000

Worked for less than 3 months - Males \

count 594.000000

mean 1338.289562

std 6127.047670

min 0.000000

25% 14.250000

50% 198.500000

75% 774.250000

max 99368.000000

Worked for less than 3 months - Females \

count 594.000000

mean 1643.340067

std 7808.832522

min 0.000000

25% 13.000000

50% 213.000000

75% 946.500000

max 122018.000000

Industrial Category - A - Cultivators - Persons \

count 594.000000

mean 865.117845

std 4274.458077

min 0.000000

25% 9.000000

50% 69.500000

75% 466.000000

max 64235.000000

Industrial Category - A - Cultivators - Males \

count 594.000000

mean 466.424242

std 2298.072295

min 0.000000

25% 5.000000

50% 35.500000

75% 244.250000

max 34632.000000

Industrial Category - A - Cultivators - Females \

count 594.000000

mean 398.693603

std 1978.682322

min 0.000000

25% 4.000000

50% 32.000000

75% 204.750000

max 29603.000000

Industrial Category - A - Agricultural labourers - Persons ... \

count 594.000000 ...

mean 12225.616162 ...

std 60458.382586 ...

min 0.000000 ...

25% 79.250000 ...

50% 1094.000000 ...

75% 6279.750000 ...

max 907752.000000 ...

Industrial Category - N to O - Females \

count 594.000000

mean 48.013468

std 222.553500

min 0.000000

25% 0.000000

50% 2.000000

75% 18.000000

max 3565.000000

Industrial Category - P to Q - Persons \

count 594.000000

mean 149.225589

std 696.553730

min 0.000000

25% 0.000000

50% 14.500000

75% 99.750000

max 11080.000000

Industrial Category - P to Q - Males \

count 594.000000

mean 54.127946

std 253.067862

min 0.000000

25% 0.000000

50% 6.000000

75% 35.750000

max 4019.000000

Industrial Category - P to Q - Females \

count 594.000000

mean 95.097643

std 444.011425

min 0.000000

25% 0.000000

50% 6.500000

75% 64.000000

max 7061.000000

Industrial Category - R to U - HHI - Persons \

count 594.000000

mean 226.707071

std 1039.953069

min 0.000000

25% 0.000000

50% 27.000000

75% 126.750000

max 16833.000000

Industrial Category - R to U - HHI - Males \

count 594.000000

mean 57.454545

std 265.230865

min 0.000000

25% 0.000000

50% 7.500000

75% 32.000000

max 4266.000000

Industrial Category - R to U - HHI - Females \

count 594.000000

mean 169.252525

std 776.206806

min 0.000000

25% 0.000000

50% 20.000000

75% 97.500000

max 12567.000000

Industrial Category - R to U - Non HHI - Persons \

count 594.000000

mean 1644.282828

std 7325.241597

min 0.000000

25% 64.500000

50% 263.500000

75% 994.000000

max 122088.000000

Industrial Category - R to U - Non HHI - Males \

count 594.000000

mean 751.528620

std 3352.811737

min 0.000000

25% 34.000000

50% 123.000000

75% 447.750000

max 55801.000000

Industrial Category - R to U - Non HHI - Females

count 594.000000

mean 892.754209

std 3988.125301

min 0.000000

25% 30.500000

50% 135.000000

75% 500.000000

max 66287.000000

[8 rows x 63 columns]

**Importance of loading and processing dataset:**

* Loading and preprocessing the dataset is an important first step inbuilding any machine learning model. However, it is especially important for house price prediction models, as house price datasets are often complex and noisy.
* By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

**1.Loading the dataset:**

*  Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.
*  The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used.However, there are some general steps that are common to most machine learning frameworks:

**a.Identify the dataset:**

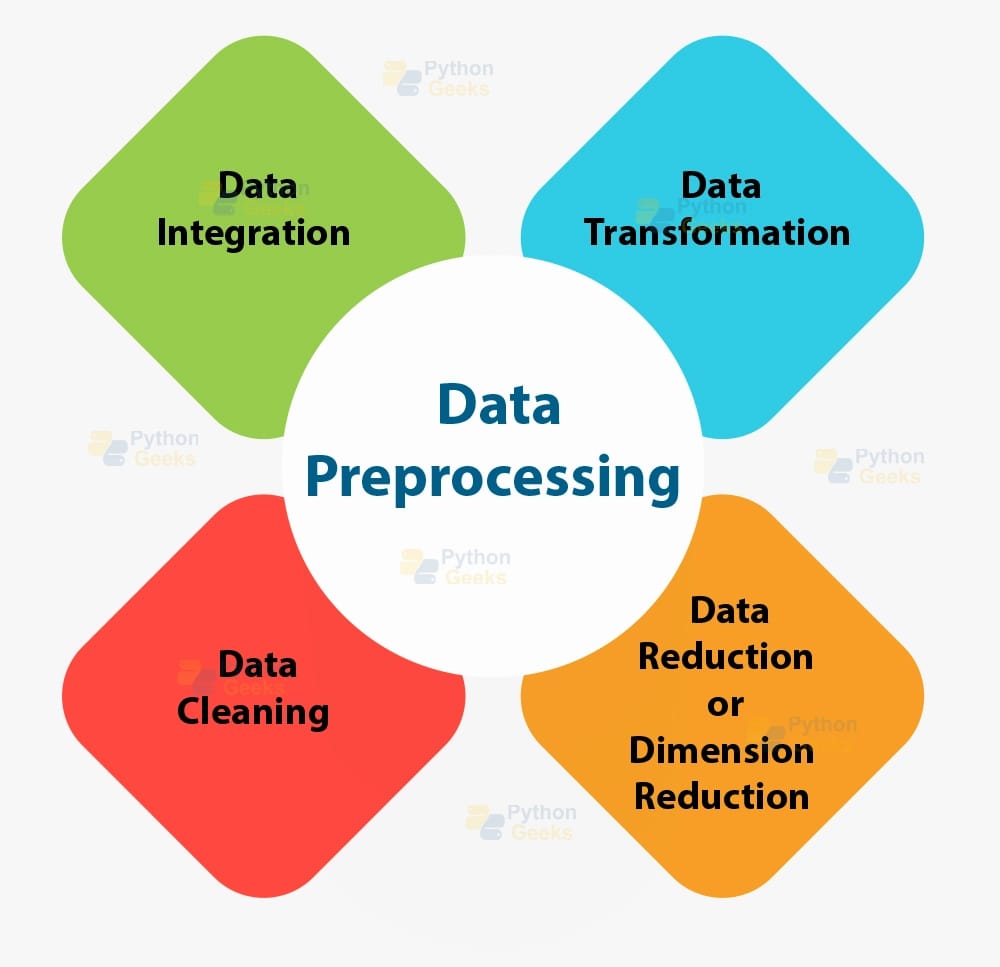
* The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

**b.Load the dataset:**

* Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

**c.Preprocess the dataset:**

* Once the dataset is loaded into the machine learning environment,you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming he data into a suitable format, and splitting the data into training and test sets



Here, how to load a dataset using machine learning in Python

**Program:**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import r2\_score,

mean\_absolute\_error,mean\_squared\_error

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Lasso

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

import xg boost as xg

%matplotlib inline

Import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_init\_.py:146:UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required forthis version of SciPy (detected version 1.23.5

warnings.warn(f"A NumPy version >={np\_minversion} and<{np\_maxversion}"

**Loading Dataset:**

dataset = pd.read\_csv(“C:\Users\ranji\Desktop\marginalworkers.csv")

**2.Preprocessing the dataset:**

* Data preprocessing is the process of cleaning, transforming, andintegrating data in order to make it ready for analysis.
* This may involve removing errors and inconsistencies, handlingmissing values, transforming the data into a consistent format, andscaling the data to a suitable range.

**Some common data preprocessing tasks include:**

* Data cleaning: This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve re moving duplicate records, correcting typos, and filling in missing values.
* Data transformation: This involves converting the data into a format that is suitable for the analysis task. For example, this may involve converting categorical data to numerical data, or scaling the data to a suitable range.
* Feature engineering: This involves creating new features from the existing data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.
* Data integration: This involves combining data from multiple sources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

Data pre processing is an essential step in many data science projects. By carefully pre processing the data, data scientists can improve the accuracy and reliability of their results.

Program:

# Importing necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

**# Step 1: Load the dataset**

data = pd.read\_csv('C:\Users\ranji\Desktop\marginalworkers.csv"csv')

**# Step 2: Exploratory Data Analysis (EDA)**

print("--- Exploratory Data Analysis ---")

print("1. Checking for Missing Values:")

missing\_values = data.isnull().sum()

print(missing\_values)

print("\n2. Descriptive Statistics:")

description = data.describe()

print(description)

**# Step 3: Feature Engineering**

print("\n--- Feature Engineering ---")

# Separate features and target variable

X = data .drop('price', axis=1)

y = data['price']

# Define which columns should be one-hot encoded (categorical)

categorical\_ cols = [' Avg. Area House Age']

# Define pre processing steps using Column Transformer and Pipeline pre processor = Column Transformer(transformers=[('num', Standard Scaler(), [' Avg. Area Number of Rooms ', ' Avg. Area Number of Bedrooms ', ' Area Population ', ' Avg. Area Income ']),('cat', One Hot Encoder(), categorical\_ cols)])

**# Step 4: Data Splitting**

print("\n--- Data Splitting ---")

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,random\_state=42)

print(f"X\_train shape: {X\_train.shape}")

print(f"X\_test shape: {X\_test.shape}")

print(f"y\_train shape: {y\_train.shape}")

print(f"y\_test shape: {y\_test.shape}")

**# Step 5: Preprocessing and Feature Scaling using Pipeline**

print("\n--- Feature Scaling ---")

model = Pipeline([('preprocessor', preprocessor),])

# Fit the preprocessing pipeline on the training data

X\_train = model.fit\_transform(X\_train)

# Transform the testing data using the fitted pipeline

X\_test = model.transform(X\_test)

print("--- Preprocessing Complete! ---")

**TEAMMATES:**

1. M KEERTHIGA
2. S RANJITHA
3. K VASUKI
4. K KAYALVIZHI