TN MARGINAL WORKERS

INTRODUCTION

Marginal workers in Tamil Nadu, as in many other parts of India, are a significant segment of the labor force. They represent individuals who are employed in various economic activities for only part of the year, often due to seasonal, casual, temporary, or underemployment reasons. Marginal workers in Tamil Nadu play a crucial role in various sectors, including agriculture, construction, and informal labor markets.

DATA PROCESSING PROCEDURE :

To create a data visualization and perform machine learning analysis on a water quality dataset using Python libraries like pandas, NumPy, matplotlib, seaborn, and scikit-learn for decision tree classification, follow these steps:

1. **Data Collection and Pre-processing**: Obtain the water quality dataset with relevant parameters Import necessary Python libraries: pandas, NumPy, matplotlib, seaborn, and scikit-learn. Load and pre-process the dataset using pandas to handle missing values, outliers, and data cleaning.

2. **Data Visualization**: Utilize matplotlib and seaborn to create visualizations of the data. Some common plots include scatter plots, histograms, and box plots to understand data distributions.

3. **Correlation Analysis**: Use pandas to calculate the correlation between different parameters in the dataset. Create correlation matrices and visualize them using heatmap plots from seaborn to identify relationships between variables.

4. **Machine Learning Preparation**: Select the target variable (e.g., marginal workers)and features (marginal workers parameters) for the machine learning model. Split the dataset into training and testing sets.

5.**Decision Tree Classifier**: Train a decision tree classifier using the Decision Tree Classifier from scikit-learn. Fit the model to the training data and evaluate its performance on the testing data.

6. **Model Evaluation**: Calculate performance metrics such as accuracy, precision, recall, and F1-score to assess the model's classification performance.

7. **Visualization of Decision Tree**: Visualize the decision tree structure using tools provided by scikit-learn, such as the plot tree function.

8. **Interpretation**: Analyse the decision tree to understand the rules and features that are most important for classification

OPERATION

- Importing python lib function and given data set:

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving DDW_B06SC_3300_State_TAMIL_NADU-2011.csv to DDW_B06SC_3300_State_TAMIL_NADU-2011.csv
```

```
[ ] sample = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.csv')
```

```
sample = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.csv')
sample.head()
```

	Table Code	State Code	District Code	Area Name	Total/ Rural/ Urban	Age group	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	...	Industrial Category - N to O - Females	Industrial Category - P to Q - Persons	Industrial Category - P to Q - Males	Industrial Category - P to Q - Females	Industrial Category - R to U - HHI - Persons	Industrial Category - R to U - HHI - Males	Industrial Category - R to U - HHI - Females	Industrial Category - R to U - Non HHI - Persons	Industrial Category - R to U - Non HHI - Males	Industria Category R to U Non HHI Female
0	B0806SC	`33	`000	State - TAMIL NADU	Total	Total	1200828	589003	611825	221386	...	3565	11080	4019	7061	16833	4266	12567	122088	55801	6628
1	B0806SC	`33	`000	State - TAMIL NADU	Total	`5-14	27791	14125	13666	2447	...	11	122	71	51	427	169	258	19305	9774	953
2	B0806SC	`33	`000	State - TAMIL NADU	Total	15-34	514340	259560	254780	92423	...	1754	7536	2718	4818	8346	2127	6219	68929	32803	3612
3	B0806SC	`33	`000	State - TAMIL NADU	Total	35-59	542581	251957	290624	99202	...	1619	3205	1131	2074	6591	1487	5104	26498	9675	1682
4	B0806SC	`33	`000	State - TAMIL NADU	Total	60+	115103	62833	52270	27165	...	175	211	93	118	1457	483	974	7065	3394	367

5 rows × 69 columns

- Identifiying number of null values and calculation mean,max,min and data types

```
sample.isnull().sum()
```

```
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
```

```
sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594 entries, 0 to 593
Data columns (total 69 columns):
 #   Column                                                    Non-Null Count  Dtype
---  ------                                                    --------------  -----
 0   Table Code                                                594 non-null    object
 1   State Code                                                594 non-null    object
 2   District Code                                             594 non-null    object
 3   Area Name                                                 594 non-null    object
 4   Total/ Rural/ Urban                                       594 non-null    object
 5   Age group                                                 594 non-null    object
 6   Worked for 3 months or more but less than 6 months -  Persons  594 non-null  int64
 7   Worked for 3 months or more but less than 6 months - Males  594 non-null  int64
 8   Worked for 3 months or more but less than 6 months - Females  594 non-null  int64
 9   Worked for less than 3 months - Persons                   594 non-null    int64
```

```
sample.describe()
```

	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	...	Industria Category N to O Female
count	5.940000e+02	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	...	594.00000
mean	1.617277e+04	7932.700337	8240.067340	2981.629630	1338.289562	1643.340067	865.117845	466.424242	398.693603	12225.616162	...	48.01346
std	7.607172e+04	36864.822704	39259.545337	13909.621137	6127.047670	7808.832522	4274.458077	2298.072295	1978.682322	60458.382586	...	222.5350
min	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.00000
25%	2.872500e+02	147.250000	144.000000	27.000000	14.250000	13.000000	9.000000	5.000000	4.000000	79.250000	...	0.00000
50%	2.225500e+03	1147.000000	1076.000000	430.000000	198.500000	213.000000	69.500000	35.500000	32.000000	1094.000000	...	2.00000
75%	9.628500e+03	4770.500000	4887.500000	1775.250000	774.250000	946.500000	466.000000	244.250000	204.750000	6279.750000	...	18.00000
max	1.200828e+06	589003.000000	611825.000000	221386.000000	99368.000000	122018.000000	64235.000000	34632.000000	29603.000000	907752.000000	...	3565.00000

8 rows × 63 columns

- Identifying null value and replacing it with mean value for accuracy

```
[10] 8 rows × 63 columns
```

```
sample.fillna(sample.mean(),inplace=True)
sample.isnull().sum()
```

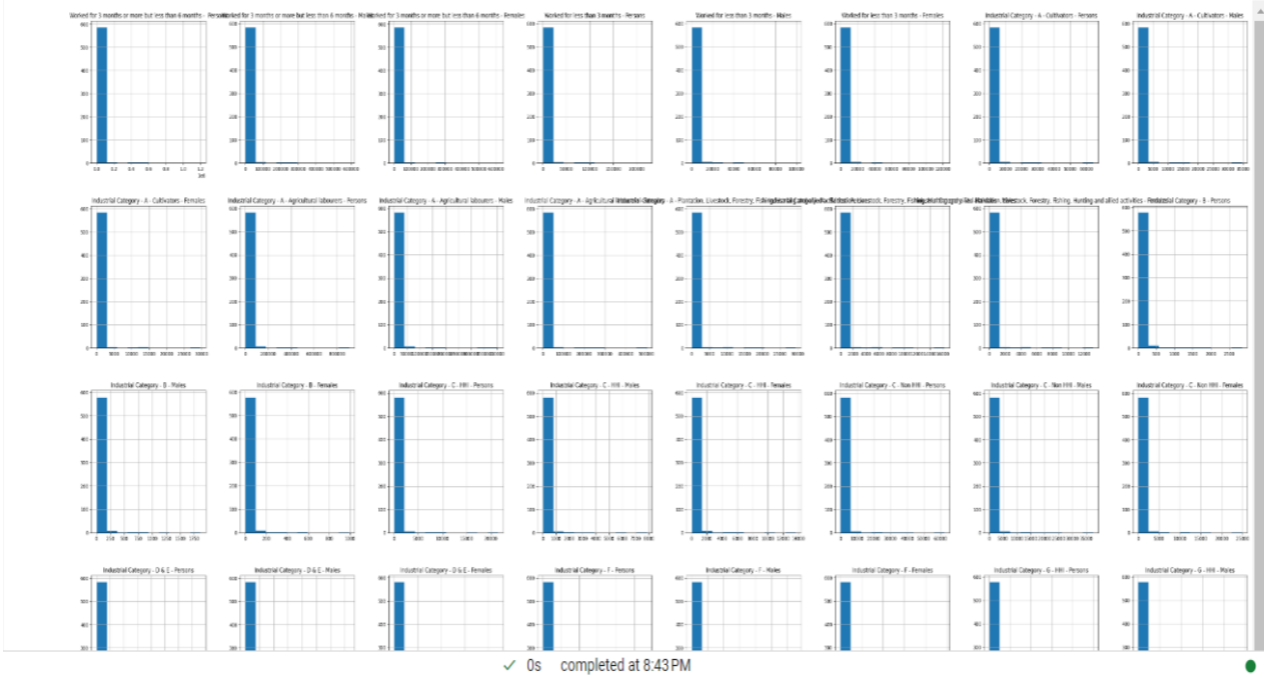
```
<ipython-input-11-37e35cec1cf1>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False
  sample.fillna(sample.mean(),inplace=True)
Table Code                                    0
State Code                                    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
```

```
sample['Table Code'].value_counts()
```

```
B0806SC    594
Name: Table Code, dtype: int64
```

- After replacement the null data set is plotted individually for analysis:

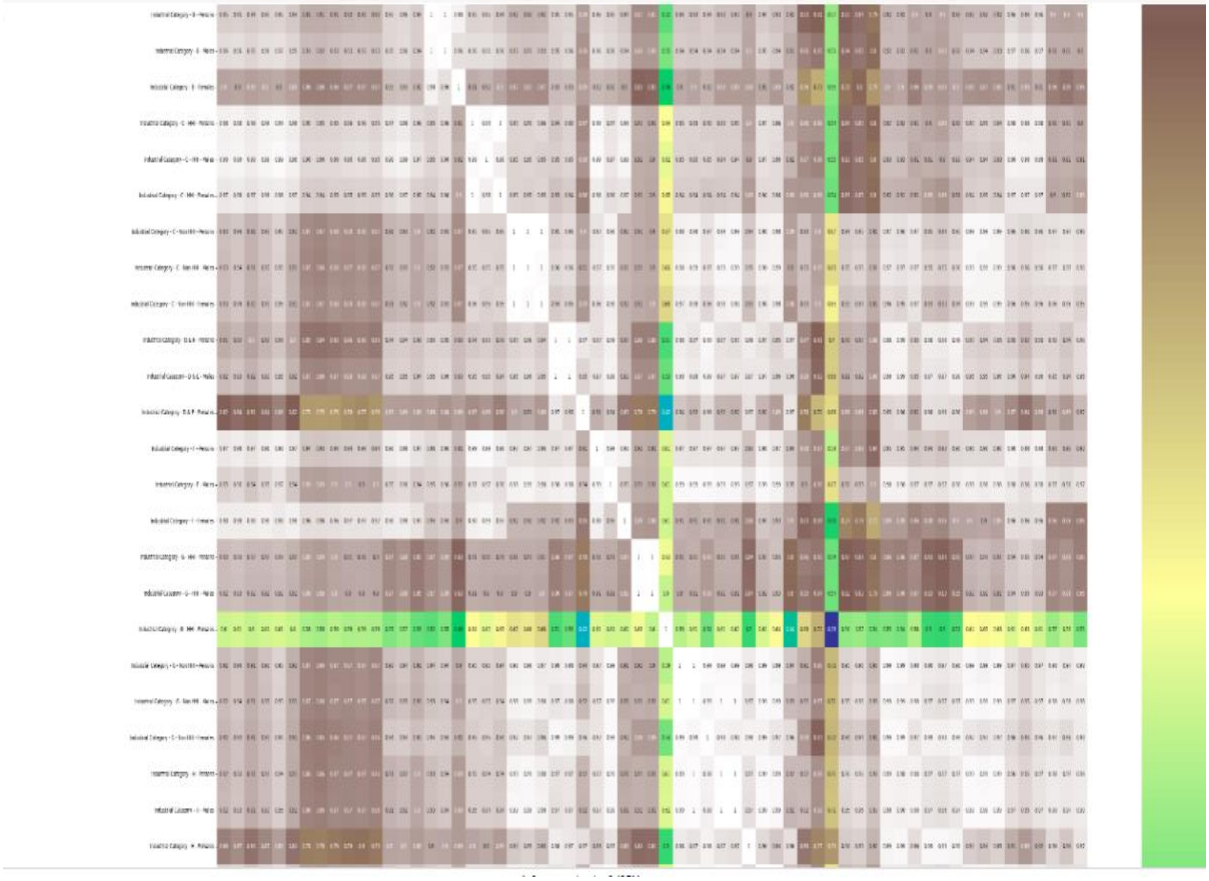
```
[ ] sample.hist(figsize=(50,50))
plt.show()
```

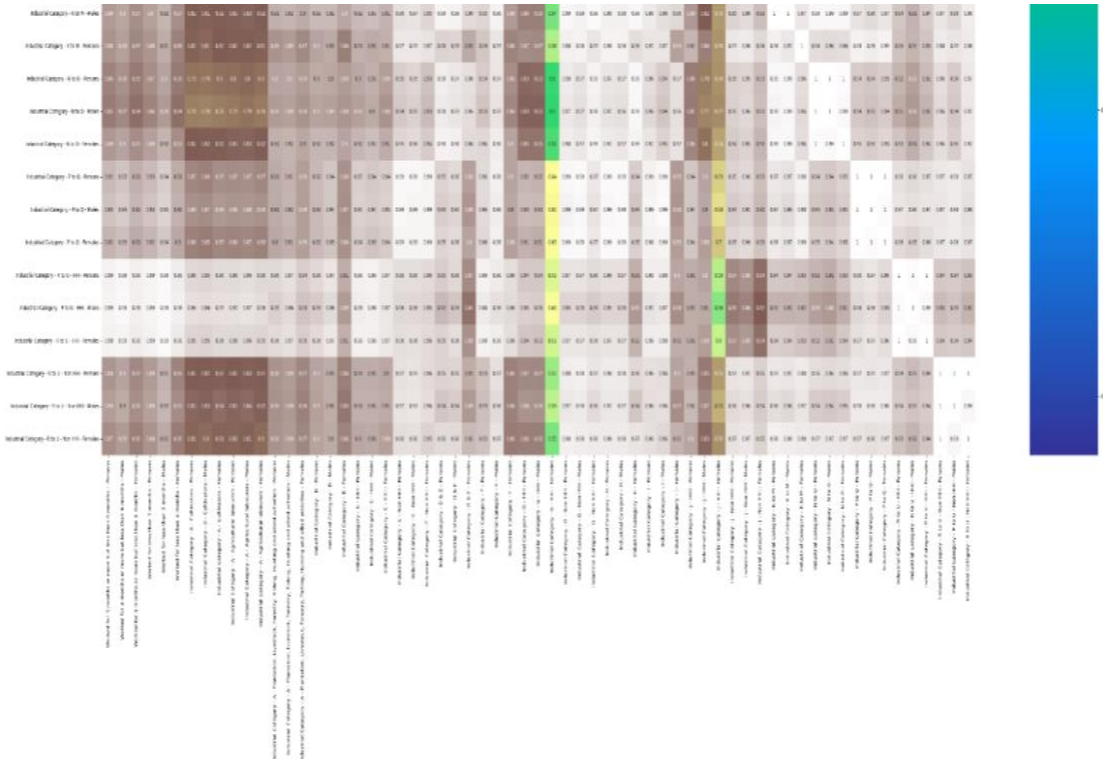


```
✓ 0s  completed at 8:43 PM
```

- Visualising the correlation of all data using function heaatmap():

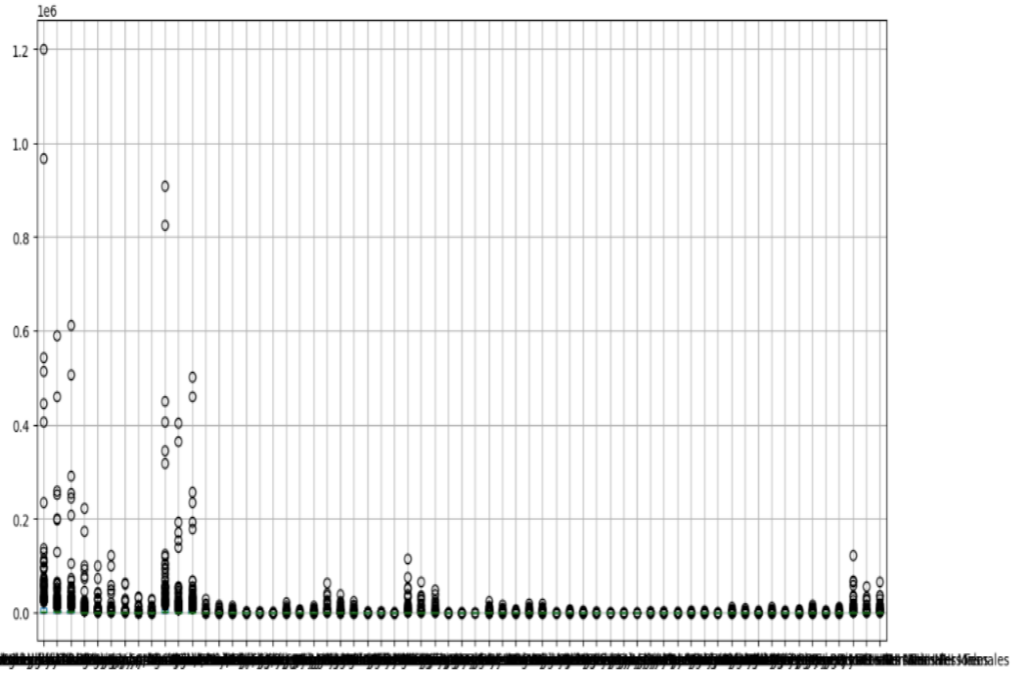
```
plt.figure(figsize=(50,63))
sns.heatmap(sample.corr(),annot=True,cmap='terrain')
plt.show
```





- Exploring the data using box plot to show the importance of not removing solid data from graph set:

<Axes: >



- Now its time to prepare the data set , divide the data set into independent and dependent

```
[ ] df = pd.get_dummies(df, columns=['Area Name'])
```

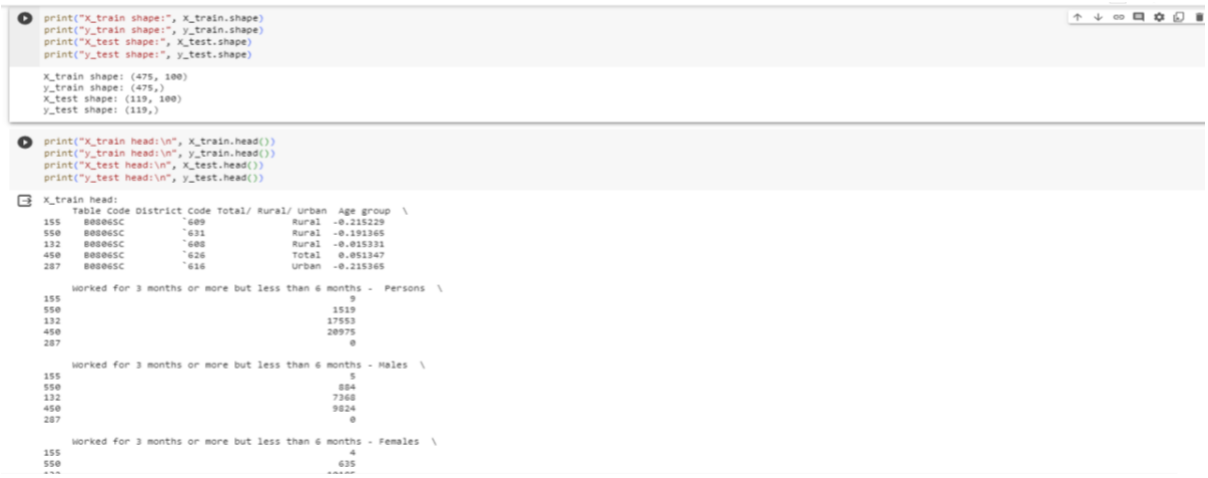
```
[ ] from sklearn.preprocessing import StandardScaler
    scaler = StandardScaler()
    df['Worked for 3 months or more but less than 6 months - Females'] = scaler.fit_transform(df[['Worked for 3 months or more but less than 6 months - Males']])
```

```
    X = df.drop('State Code', axis=1)
    y = df['State Code']
```

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[ ] print("X_train shape:", X_train.shape)
    print("y_train shape:", y_train.shape)
    print("X_test shape:", X_test.shape)
    print("y_test shape:", y_test.shape)
```

```
X_train shape: (475, 100)
y_train shape: (475,)
X_test shape: (119, 100)
y_test shape: (119,)
```



```
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
print("X_test shape:", X_test.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (475, 100)
y_train shape: (475,)
X_test shape: (119, 100)
y_test shape: (119,)
```

```
print("X_train head:\n", X_train.head())
print("y_train head:\n", y_train.head())
print("X_test head:\n", X_test.head())
print("y_test head:\n", y_test.head())
```

```
X_train head:
     Table Code District Code Total/ Rural/ Urban  Age group  \
155    B08065C          `609              Rural  -0.215229
550    B08065C          `631              Rural  -0.191365
132    B08065C          `608              Rural  -0.015331
450    B08065C          `626              Total   0.051347
287    B08065C          `616              Urban  -0.215365

     Worked for 3 months or more but less than 6 months -  Persons  \
155                                                  9
550                                               1519
132                                              17553
450                                              20975
287                                                  0

     Worked for 3 months or more but less than 6 months - Males  \
155                                                  5
550                                                884
132                                               7368
450                                               9824
287                                                  0

     Worked for 3 months or more but less than 6 months - Females  \
155                                                  4
550                                                635
```

CONCLUSION

In conclusion, data analytics in the context of Tamil Nadu (TN) marginal workers serves as a valuable tool for gaining insights into the characteristics, challenges, and economic impact of this specific group of laborers. By analyzing data related to marginal workers, policymakers, researchers, and organizations can make informed decisions to enhance the well-being of these workers and strengthen the labor market. Some key takeaways from the objectives of data analytics in this area include:

1. Understanding employment patterns and identifying vulnerable groups are essential for targeted interventions and support.

2. Seasonal variations in marginal employment should be considered in planning programs and policies.

3. Evaluating economic impact and monitoring labor market trends are crucial for informed decision-making.

4. Assessing the effectiveness of policies and forecasting future labor force dynamics helps in achieving desired outcomes.

5. Policy formulation should be data-driven and tailored to the specific needs of marginal workers.

6. Identifying barriers to full employment is vital for creating pathways to stable, year-round work.

7. Data analytics can also contribute to improving data collection methods for more accurate information.

In essence, data analytics is a powerful tool for shedding light on the dynamics of TN marginal workers and guiding efforts to improve their economic stability and well-being. It aids in developing evidence-based policies and programs that can bring positive changes to the lives of marginal workers and promote inclusive economic growth in the region.