TN MARGINAL WORKERS

Development part 1:

Introduction:

The Optimizing project success through Best practices for loading and Preprocessing marginal worker Datasets in Tamil Nadu. Today, we'll Explore the challenges of working with Marginal worker datasets and how to Overcome them.

Defifining Marginal Worker Datasets:

Marginal worker datasets are composed Of individuals who work for less than six Months out of the year. These datasets Are often incomplete and require careful Preprocessing to ensure accuracy. In This section, we'll explore the unique Challenges of working with marginal Worker data and how to overcome Them. Preprocessing marginal worker data Datasets involves transforming raw Data into a usable format. In this Section, we'll explore best practices for Preprocessing marginal worker data, Including data normalization, outlier Detection, and feature selection.

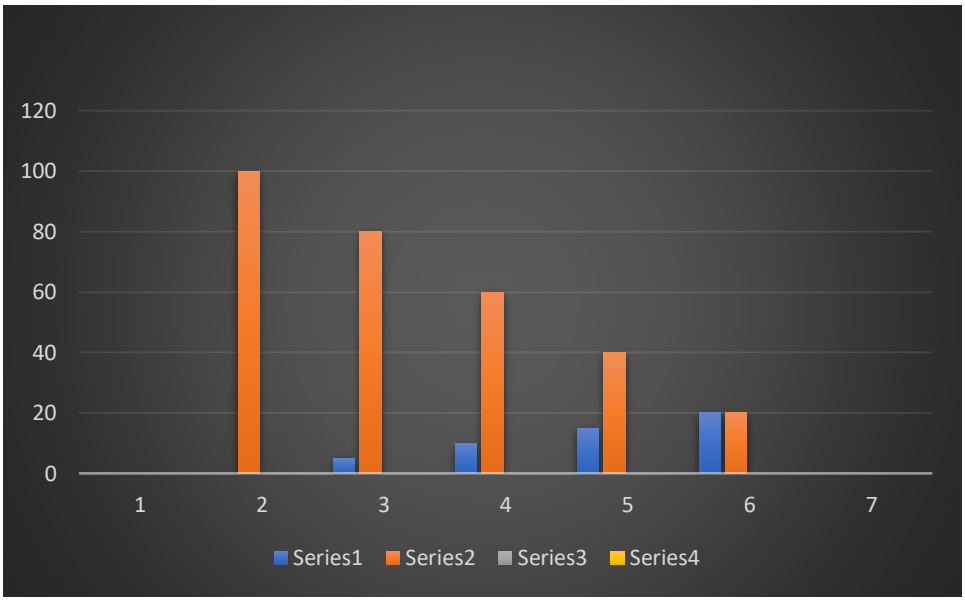
Best Practices for Loading Marginal Worker:

Datasets Best Practices for Loading Marginal Worker Datasets Loading marginal worker datasets Requires careful attention to detail. In This section, we'll explore best Practices for loading and cleaning Marginal worker data, including Verifying data sources, identifying Missing data, and performing quality Checks.

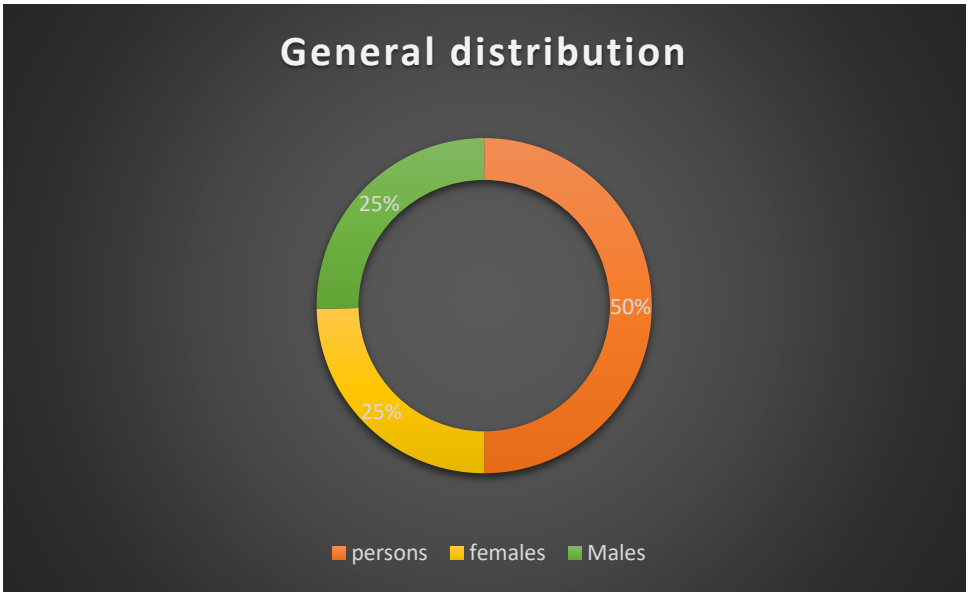
Analyzing marginal worker datasets:

Analyzing marginal worker datasets requires Specialized techniques to account for missing And incomplete data. In this section, we'll explore Best practices for analyzing marginal worker Data, including imputation, regression analysis,And machine learning techniques.

```
Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns


# Load the dataset

Data = pd.read_csv("/MARGINAL_WORKERS.csv")

Data.dropna(inplace=True)


# Demographic Analysis

# Age Distribution

Plt.figure(figsize=(20, 10))

Sns.histplot(data['Age group'], bins=20, kde=True)

Plt.title("Age Distribution of Marginal Workers in Tamil Nadu")

Plt.xlabel("Age Group")

Plt.ylabel("Count")

Plt.xticks(rotation=45)

Plt.show()
```



```
gender_counts = data[['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']].sum()

plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=90)
plt.title("Gender Distribution of Marginal Workers in Tamil Nadu")
plt.show()
```



Python offers several powerful libraries for data manipulation, some of the popular ones include:

- **NumPy:**
  A fundamental package for scientific computing with Python, which provides support for arrays, matrices, and mathematical functions.

- Pandas:
  A fast, powerful, and flexible open-source data analysis and manipulation tool, built on top of NumPy.

- Matplotlib:
  A widely-used plotting library that provides publication-quality 2D plotting as well as simple 3D plotting.

- Seaborn:
  A statistical data visualization library that provides a high-level interface for drawing attractive and informative statistical graphics.

- SciPy:
  A Python-based ecosystem of open-source software for mathematics, science, and engineering. It includes modules for optimization, linear algebra, integration, interpolation, special functions, and more.

- Scikit-learn:

  A simple and efficient tool for data mining and data analysis, built on NumPy, SciPy, and Matplotlib. It is used for tasks such as classification, regression, and clustering.

Pandas is a widely used open-source data analysis and manipulation library for Python. It provides data structures and functions that make working with structured data straightforward. Some key features of Pandas include:

1.DataFrame object for data manipulation with integrated indexing.

2.Tools for reading and writing data between in-memory data structures and different file formats.

3. Data alignment and handling of missing data.

4. Reshaping and pivoting of data sets.

5. Label-based slicing, fancy indexing, and subsetting of large data sets.

6. Built-in data visualization features.

Pandas is particularly well-suited for tasks such as data cleaning, preparation, and analysis. It's commonly used in data science and machine learning workflows.

Conclusion:

In conclusion, optimizing project success through best Practices for loading and preprocessing marginal worker Datasets in Tamil Nadu requires careful attention to detail and Specialized techniques. By following the best practices Outlined in this presentation, you can ensure the accuracy and Reliability of your data and improve project outcomes.