

## **PHASE 3 ASSIGNMENT**

**PROJECT TITLE:** PREPROCESSING THE DATASET

### **PROBLEM STATEMENT:**

AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of Companies (RoC).

### **PROBLEM DEFINITION:**

The problem is to perform an AI-driven exploration and predictive analysis on the master details of companies registered with the Registrar of Companies (RoC). The objective is to uncover hidden patterns, gain insights into the company landscape, and forecast future registration trends. This project aims to develop predictive models using advanced Artificial Intelligence techniques to anticipate future company registrations and support informed decision-making for businesses, investors, and policymakers.

**GITHUB LINK:** <https://github.com/Sheebha/RoC.git>

<https://github.com/Sheebha/innovation.git>

### **DOCUMENT:**

Building the project by preprocessing the data.

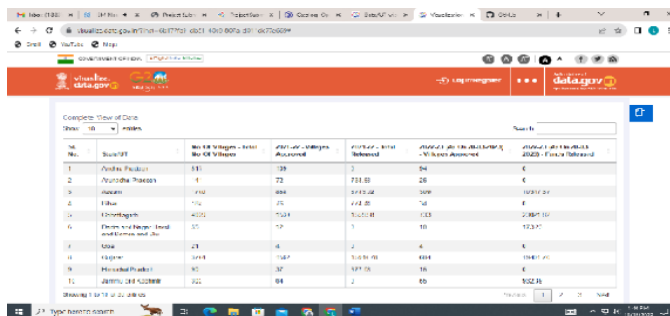
**DATASET LINK:** <https://tn.data.gov.in/resource/company-master-data-tamil-nadu-upto-28th-february-2019>

Preprocessing a dataset is a crucial step in preparing data for machine learning models. The specific steps can vary depending on the nature of your data and the problem you're trying to solve. However, here's a general set of steps you might follow:

## 1. **\*\*Import Libraries:\*\***

- Import the necessary libraries for data manipulation and analysis such as Pandas, NumPy, and others.

```
```python
import pandas as pd
import numpy as np
```
```



The screenshot shows the Kaggle Datasets browser interface. At the top, there's a search bar and navigation links. Below, a table titled 'Complete View of Data' is displayed. The table has columns for 'No.', 'Dataset', 'Size', 'Downloads', 'Views', 'Created', and 'Updated'. The data rows are as follows:

| No. | Dataset                                     | Size | Downloads | Views  | Created | Updated |
|-----|---|------|-----------|--------|---------|---------|
| 1   | Avocado Production                          | 1.1M | 139       | 34     | 6       | 6       |
| 2   | Avocado Prices                              | 1.1M | 72        | 28     | 6       | 6       |
| 3   | Avocado                                     | 1.1M | 88        | 113.48 | 109     | 109.17  |
| 4   | Blue  | 1.1M | 25        | 1.1M   | 1.1M    | 1.1M    |
| 5   | Cherry Apples                               | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |
| 6   | Cherry and Apple - Small and Large and etc. | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |
| 7   | Cherry                                      | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |
| 8   | Cherry                                      | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |
| 9   | Cherry                                      | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |
| 10  | Cherry                                      | 1.1M | 1.1M      | 1.1M   | 1.1M    | 1.1M    |

## 2. **\*\*Load the Dataset:\*\***

- Read the dataset into a Pandas DataFrame.

```
```python
data = pd.read_csv('your_dataset.csv')
```
```



### 3. \*\*Explore the Data:\*\*

- Check for missing values, understand the structure of the data, and explore basic statistics.

```
```python
```

```
# Check for missing values
```

```
print(data.isnull().sum())
```

```
# Basic statistics
```

```
print(data.describe())
```



#### 4. \*\*Handle Missing Values:\*\*

- Decide on a strategy for handling missing data. Options include dropping missing values, filling them with mean or median, or using more advanced imputation techniques.

```
```python
```

```
# Drop rows with missing values
```

```
data = data.dropna()
```

```
# Fill missing values with mean
```

```
data = data.fillna(data.mean())
```

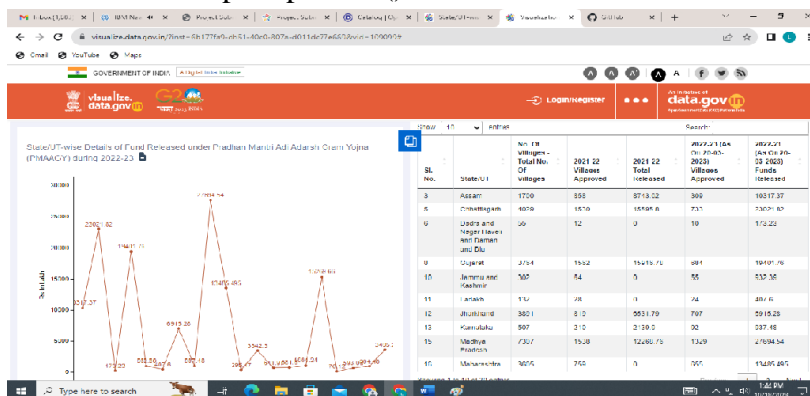
```
```
```

#### 5. \*\*Remove Duplicates:\*\*

- Check for and remove duplicate rows.

```
```python
```

```
data = data.drop_duplicates()
```



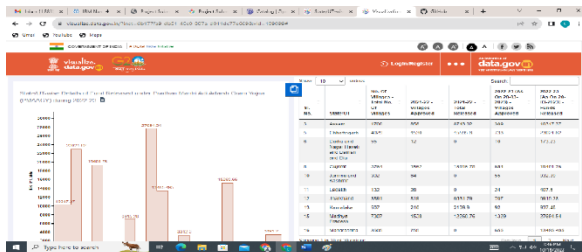
## 6. **\*\*Handle Categorical Data:\*\***

- Convert categorical variables into numerical format, using techniques like one-hot encoding or label encoding.

```
```python
# One-hot encoding

data = pd.get_dummies(data, columns=['categorical_column'])

```
```



## 7. **\*\*Feature Scaling:\*\***

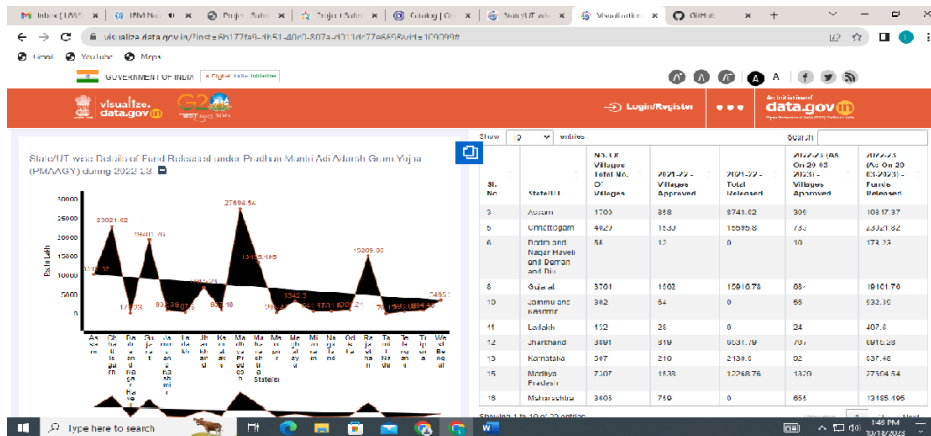
- Standardize or normalize numerical features to ensure they are on similar scales.

```
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data[['numerical_column']] = scaler.fit_transform(data[['numerical_column']])

```
```



## 8. \*\*Feature Engineering:\*\*

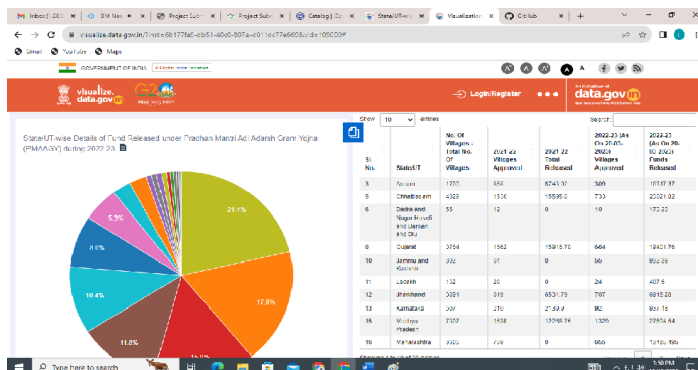
- Create new features or transform existing ones to better represent the underlying patterns in the data.

```
python
```

```
# Example: Create a new feature
```

```
data['new_feature'] = data['feature1'] * data['feature2']
```

```
'''
```



## 9. \*\*Split the Dataset:\*\*

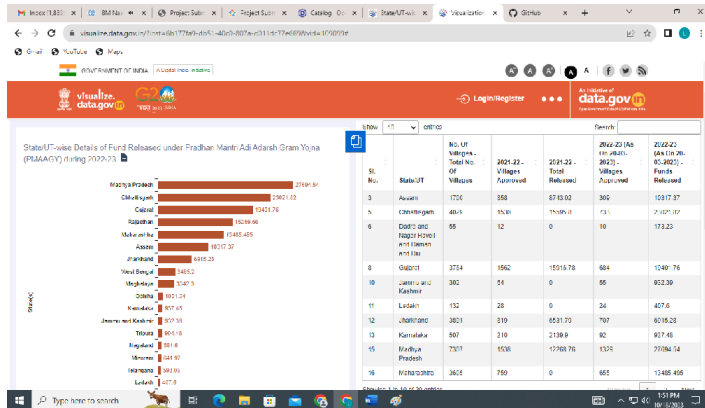
- Split the dataset into training and testing sets.

```
```python
```

```
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)
```

```
```
```

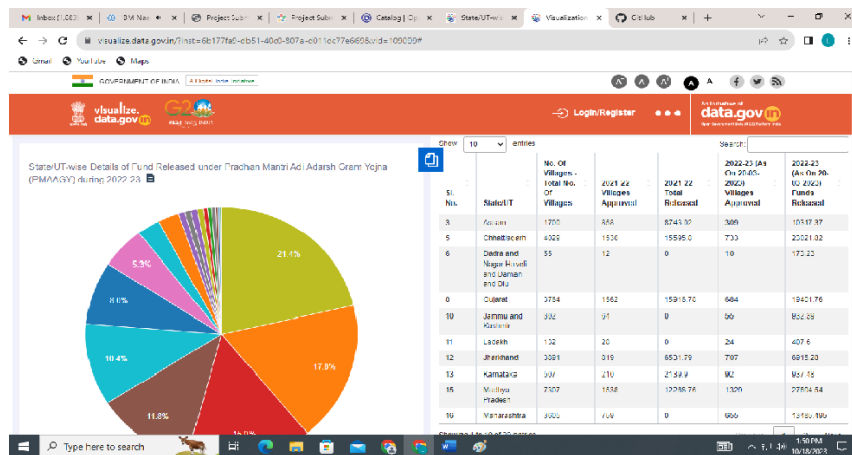


## 10. \*\*Save Preprocessed Data (Optional):\*\*

- Save the preprocessed data to a new file for future use.

```
```python
```

```
data.to_csv('preprocessed_data.csv', index=False)
```



**SUBMITTED BY,**

**STUDENT REG NO: 711221104050**

**NAAN MUDHALVAN: au711221104050**