Marginal Workers in Tamil Nadu

Introduction

This section serves as the foundation for understanding the purpose and scope of analyzing data on marginal workers in Tamil Nadu. Dive into the significance of this dataset, exploring the socio-economic implications and potential policy implications. Gain a holistic worldview that goes beyond the numbers.

Data Exploration and Visualization:

Data Collection: Obtain the dataset containing information about marginal workers. This dataset may include various features like age, gender, education, location, income, etc.

Data Preprocessing:

- Handle missing values if any.
- Encode categorical variables if necessary.
- Normalize or standardize numerical features.

Exploratory Data Analysis (EDA):

- Summarize the basic statistics of the dataset.
- Visualize the distribution of different features using histograms, box plots, and scatter plots.
- Examine correlations between features using a correlation matrix.

Understand the Target Variable:

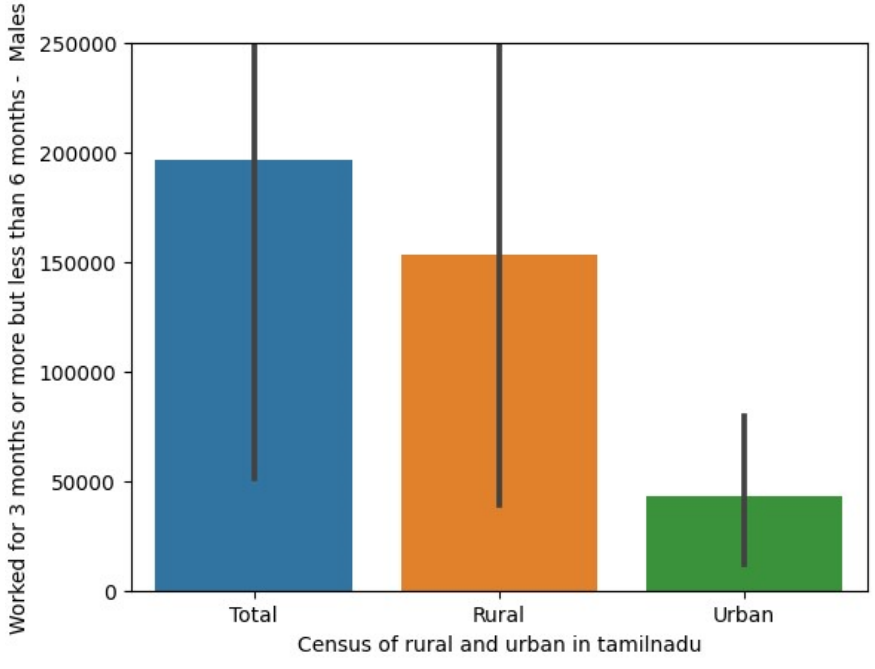
- Identify and understand the target variable (e.g., whether a person is a marginal worker or not).
- Explore the distribution of the target variable to check for class imbalances.

Data Visualization:

- Create meaningful visualizations like bar charts, pie charts, or heatmaps to better understand the data and relationships between variables.
- Visualize the distribution of marginal workers within different categories (e.g., age groups, education levels, regions, etc.).

Visualizing Dataset

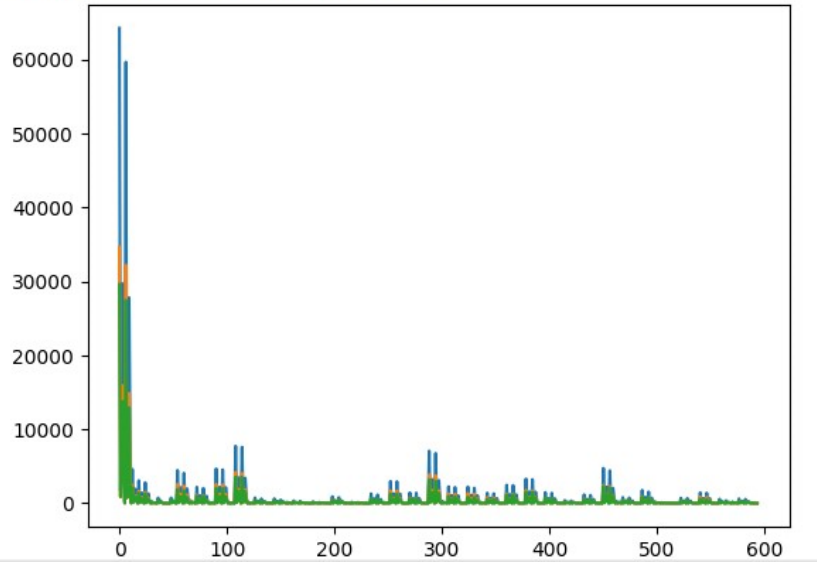
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')
x = df['Total/ Rural/ Urban'].loc[0:17]
y =df['Worked for 3 months or more but less than 6 months - Males'].loc[0:17]
sns.barplot(x=x, y=y)
plt.xlabel('Census of rural and urban in tamilnadu')
plt.ylabel('Worked for 3 months or more but less than 6 months -  Males')
plt.ylim(0, 250000)
plt.show()
```



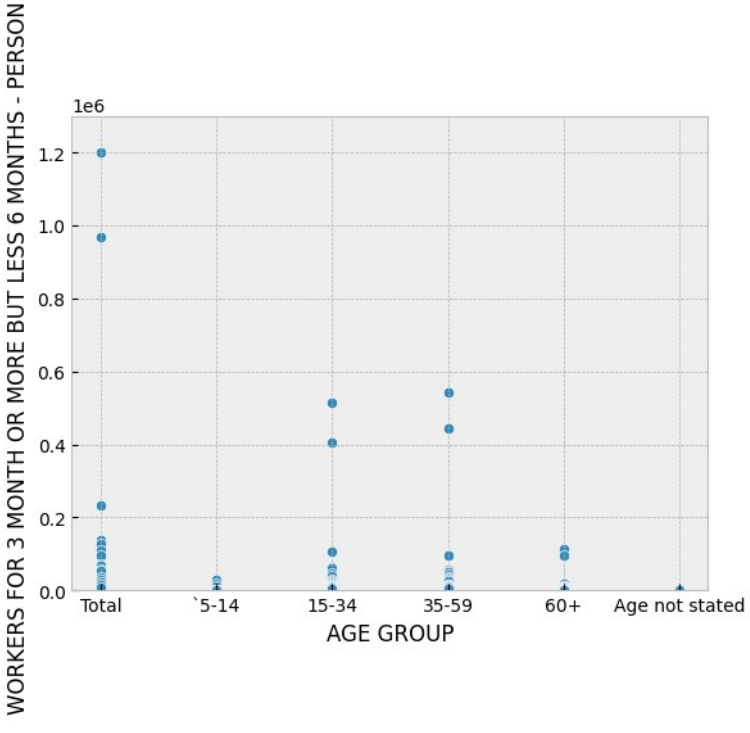
Visualizing data related to working periods and the rural-urban census in Tamil Nadu, particularly in the context of marginal workers, can provide valuable insights into the state's labor dynamics

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')
a = df[['Industrial Category - A - Cultivators - Persons','Industrial Category - A -
Cultivators - Males','Industrial Category - A - Cultivators - Females']]
plt.plot(a)
```

```
[<matplotlib.lines.Line2D at 0x2f8aefc5b20>,
 <matplotlib.lines.Line2D at 0x2f8aefc5b80>,
 <matplotlib.lines.Line2D at 0x2f8aefc5ca0>]
```




```
import pandas as pd
import seaborn as sns
df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')
x = df['Age group']
y =df['Worked for 3 months or more but less than 6 months -  Persons']
sns.scatterplot(x=x, y=y)
plt.xlabel('AGE GROUP')
plt.ylabel('WORKERS FOR 3 MONTH OR MORE BUT LESS 6 MONTHS - PERSON ')
plt.ylim(0, 1300000)

plt.show()
```



Visualizing data related to specific industrial categories within the context of marginal workers in Tamil Nadu, such as 'Cultivators,' allows us to delve into the gender-specific dynamics of this occupation. These categories, namely 'Cultivators - Persons,' 'Cultivators -

Males,' and 'Cultivators - Females,' shed light on the composition of the agricultural workforce in the state.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')
x = np.array(df['Worked for 3 months or more but less than 6 months -  Persons'])
y = np.array(df['Worked for less than 3 months - Persons'])
plt.xlabel('workers for 3 months or more but less than 6 months - persons')
plt.ylabel('workers for less than 3 months - person')
plt.plot(x,y)
plt.show()
```

Supervised Learning:

Split the Data:

- Split the dataset into training and testing subsets to evaluate the performance of the supervised learning model.

Select Features:

- Identify the most relevant features for predicting whether a person is a marginal worker. Feature selection techniques such as feature importance or recursive feature elimination can be used.

Choose a Supervised Learning Algorithm:

- Select an appropriate supervised learning algorithm for classification tasks. Common choices include:
  - Logistic Regression
  - Decision Trees
  - Random Forest
  - Support Vector Machines
  - Naive Bayes

- Gradient Boosting methods (e.g., XGBoost, LightGBM)
- Neural Networks (Deep Learning)

Model Training:

- Train the selected model on the training data. Tune hyperparameters to optimize model performance.

Model Evaluation:

- Evaluate the model on the testing dataset using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score, ROC AUC).

Interpretability:

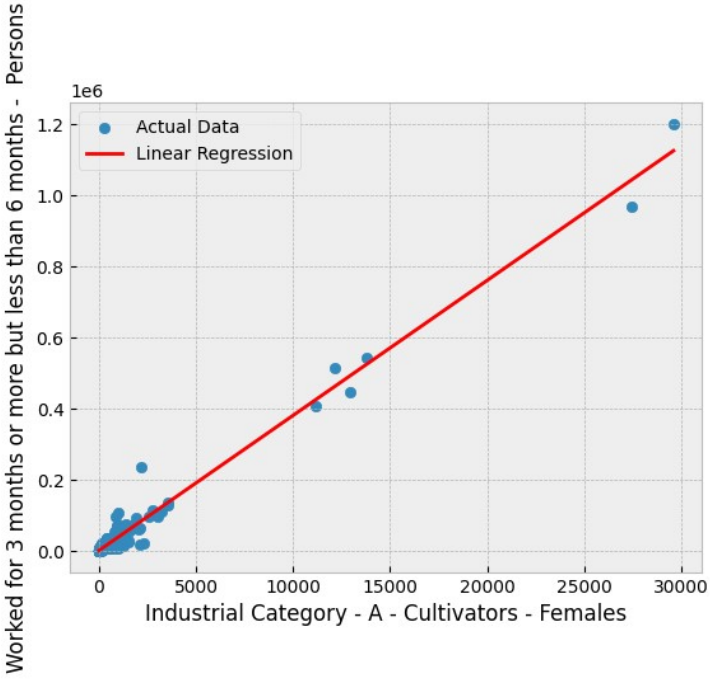
- Analyze the model's predictions and decision boundaries to understand which features contribute most to predicting marginal workers.

Visualization:

- Visualize the model's performance using ROC curves, confusion matrices, and other relevant plots.

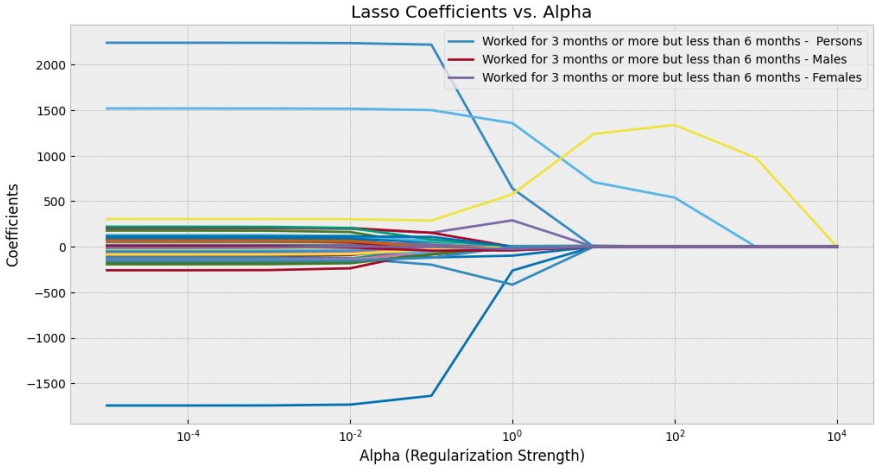
LINEAR REGRESSION

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')
X =np.array( df['Industrial Category - A - Cultivators - Females']).reshape(-1,1)
y = np.array(df['Worked for 3 months or more but less than 6 months -  Persons'])

# Create a linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(X, y)

# Make predictions
y_pred = model.predict(X)

# Plot the results
plt.scatter(X, y, label='Actual Data')
plt.plot(X, y_pred, color='red', label='Linear Regression')
plt.xlabel('Industrial Category - A - Cultivators - Females')
plt.ylabel('Worked for 3 months or more but less than 6 months -  Persons')
plt.legend()
plt.show()
```



Linear regression is a fundamental statistical technique commonly used in data analytics to model and understand the relationship between a dependent variable (also known as the target or outcome variable) and one or more independent variables (predictors or features). It is especially useful when trying to predict or explain a continuous numeric outcome

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import Lasso
from sklearn.preprocessing import StandardScaler

# Load your dataset (replace 'your_dataset.csv' with the actual dataset path)
data = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')

# Assume 'X' contains your feature variables, and 'y' contains the target variable
X = data.drop(['Table Code','State Code','District Code','Area Name','Total/ Rural/ Urban','Age group'], axis=1)
y = data['Industrial Category - A - Cultivators - Females']

# Standardize the feature variables (mean=0, variance=1)
scaler = StandardScaler()
X = scaler.fit_transform(X)

alphas = np.logspace(-5, 4, 10)  # Vary alpha from 0.0001 to 10000

coefs = []
for alpha in alphas:
    lasso_reg = Lasso(alpha=alpha)
    lasso_reg.fit(X, y)
    coefs.append(lasso_reg.coef_)

# Create a plot
plt.figure(figsize=(12, 6))
plt.plot(alphas, coefs)
plt.xscale('log')
plt.xlabel('Alpha (Regularization Strength)')
plt.ylabel('Coefficients')
plt.title('Lasso Coefficients vs. Alpha')
plt.axis('tight')
plt.legend(['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 mo
plt.show()
```



CONCLUSION

In conclusion, the exploration, visualization, and application of supervised learning algorithms to marginal workers' data is a valuable and multi-faceted process that yields critical insights and informs decision-making. the combination of data exploration and visualization with supervised learning algorithms empowers us to make informed decisions and create targeted solutions for the betterment of marginalized workers