**Title:**

TN Marginal Workers Assessment

**Abstract:**

This study undertakes a comprehensive socioeconomic analysis of marginal workers in the state of Tamil Nadu, India, with a focus on their demographic characteristics, including age, industrial category, and sex. Marginal workers play a critical role in the labor force, often facing unique challenges and disparities. Understanding their demographics and socio-economic status is essential for informed policymaking and interventions.

The research employs a data-driven approach, collecting and analyzing data from various sources, including government databases, surveys, and research institutions. Demographic attributes such as age, industrial category, and sex are scrutinized to reveal intricate patterns, disparities, and trends among marginal workers.

To enhance the accessibility of the findings, the study employs data visualization techniques. Bar charts, pie charts, and heatmaps are generated to represent the distribution of marginal workers across different categories. These visualizations help in conveying complex demographic insights in a comprehensible manner.

Through this analysis, the project aims to shed light on the socio-economic landscape of marginal workers in Tamil Nadu. The findings are expected to offer valuable insights for policymakers, researchers, and organizations working toward the improvement of the lives and livelihoods of marginal workers in the region. This study underscores the significance of data-driven decision-making in addressing socio-economic disparities and fostering inclusive growth.

Top of Form

**Introduction:**

Marginal workers constitute a significant segment of the labor force in Tamil Nadu, India, and are integral to various economic activities. These individuals often find themselves in precarious employment, with limited access to social security, education, and healthcare. The challenges faced by marginal workers are manifold, and their socio-economic conditions reflect the broader complexities of the labor market in the region.

This study embarks on a comprehensive socioeconomic analysis with a central focus on the demographic characteristics of marginal workers in Tamil Nadu. By scrutinizing their age, industrial category, and sex, we aim to unravel the intricate tapestry that defines this vital workforce. Understanding the demographic dimensions of marginal workers is crucial, as it provides a foundation for informed policymaking, targeted interventions, and advocacy to improve their conditions.

A data-driven approach underpins our research methodology, drawing from a rich tapestry of sources that include government databases, surveys, and contributions from esteemed research institutions. These sources collectively form the basis for a holistic examination of the socio-economic landscape inhabited by marginal workers.

By delving into the demographic attributes, our analysis seeks to uncover the subtle patterns, disparities, and evolving trends that shape the lives of these workers. The interplay between age, industrial category, and gender unveils a nuanced understanding of their struggles and opportunities.

To make our findings more accessible and impactful, we leverage data visualization techniques. Bar charts, pie charts, and heatmaps are instrumental in simplifying the presentation of complex demographic insights. These visualizations offer a medium for stakeholders, researchers, and policymakers to comprehend the multifaceted dynamics of marginal workers, transcending data into actionable knowledge.

The overarching goal of this project is to illuminate the socio-economic landscape inhabited by marginal workers in Tamil Nadu. Through the analytical lens, we seek to generate insights that inform the decisions of policymakers and inspire initiatives aimed at enhancing the quality of life and livelihoods of marginal workers. In doing so, we underscore the pivotal role of data-driven decision-making in addressing socio-economic disparities and in advancing the cause of inclusive growth. This study is a testament to our commitment to empower the marginalized and promote socio-economic equity in the region.

**The Need for Assessment for Marginal Workers:**

1. **Understanding Marginalized Workforce**

2. **Informed Policymaking**:

3. **Identifying Disparities**

4. **Data-Driven Decisions**

5. **Visual Communication**

6. **Targeted Interventions**

**7.** **Policy Evaluation**

**Importance in the Business Context:**

1. **Workforce Diversity and Inclusion**: Businesses are increasingly recognizing the importance of a diverse and inclusive workforce. Understanding the demographics of marginalized workers helps organizations assess their own diversity and inclusion efforts and identify areas for improvement.
2. **Tailored Human Resource Policies**: Businesses can use the insights from the analysis to design more targeted human resource policies. For instance, they can adapt hiring, training, and career development programs to better suit the needs of marginalized workers.
3. **Market Segmentation**: Companies can use demographic data to refine their market segmentation strategies. By understanding the characteristics of marginalized workers, they can tailor products and services to cater to this specific consumer group.
4. **Social Responsibility**: Demonstrating a commitment to social responsibility is essential for many businesses. An analysis of marginalized worker demographics can lead to initiatives aimed at improving the lives of these workers, which can enhance a company's reputation and brand.
5. **Supply Chain Management**: Understanding the demographic characteristics of workers in different industrial categories is valuable for companies managing supply chains. It can help in identifying potential labor-related risks and opportunities in the supply chain.
6. **Employee Engagement**: Knowing the demographics of the workforce can lead to more effective employee engagement strategies. It helps in creating a work environment that respects the unique needs and challenges faced by marginalized workers.
7. **Compliance and Labor Laws**: Companies need to adhere to labor laws and regulations. Analyzing worker demographics can assist in ensuring compliance and taking proactive steps to address potential legal issues.
8. **Talent Acquisition**: An analysis of marginalized worker demographics can help in recruitment efforts. Companies can target underrepresented groups, tapping into a wider talent pool, which is important in a competitive job market.
9. **Competitive Advantage**: Businesses that proactively address the needs of marginalized workers can gain a competitive advantage. They can attract talent that may be overlooked by other firms, leading to a more diverse and innovative workforce.
10. **Long-Term Sustainability**: Prioritizing marginalized worker demographics aligns with the principles of sustainable and responsible business practices. By addressing social and economic disparities, companies can contribute to a more sustainable future.

**Specific Goal:**

1. **Data Collection**: Collect relevant data on marginal workers in Tamil Nadu, including age, industrial category, and sex, from various sources, such as government databases, surveys, and research institutions.
2. **Data Preprocessing**: Clean and preprocess the collected data to ensure accuracy and consistency. Handle missing values and outliers if necessary.
3. **Demographic Analysis**: Analyze the demographic characteristics of marginal workers, including age distribution, gender distribution, and the distribution of workers across different industrial categories.
4. **Identify Disparities**: Determine if there are significant disparities or trends within the data. For example, identify if certain age groups or industrial categories are more prominent among marginal workers.
5. **Visualization Development**: Create visualizations to represent the demographic distribution of marginal workers. This can include bar charts, pie charts, and heatmaps to effectively convey the insights.
6. **Interpretation**: Interpret the visualizations to extract meaningful insights about the socio-economic characteristics of marginal workers in Tamil Nadu. Understand the implications of the data.
7. **Policy Recommendations**: Based on the analysis and insights, propose policy recommendations or interventions to address any disparities or challenges faced by marginalized workers.
8. **Stakeholder Engagement**: Engage with relevant stakeholders, including policymakers, NGOs, and advocacy groups, to share the findings and recommendations for potential collaboration.
9. **Documentation and Reporting**: Prepare a comprehensive report summarizing the analysis, visualization, and insights. Ensure that the report is accessible to a broad audience, including non-technical stakeholders.
10. **Knowledge Sharing**: Share the results and findings through presentations, seminars, or public awareness campaigns to create a greater understanding of the socio-economic conditions of marginal workers.
11. **Continuous Monitoring**: If feasible, set up a system for continuous data monitoring and periodic updates to track changes in the demographic characteristics and evaluate the impact of any implemented interventions over time.
12. **Ethical Considerations**: Ensure that all aspects of the project, including data collection, handling, and reporting, adhere to ethical standards and respect the privacy and rights of the subjects involved.

**What We Aim to Achieve:**

**1. Data-Driven Understanding**

**2. Identification of Disparities**

**3. Informed Policymaking**

**4. Data Visualization for Accessibility**

**5. Advocacy and Awareness:**

**6. Policy Recommendations**

**7. Impact Assessment**

**8. Empowerment**

**Dataset Description:**

The dataset used for Assessment of Marginal Workers with machine learning project is a collection of historical sales and product-related information. It provides details on the industrial workers over a specific time period. The dataset contains the following key features:

1. Age

2. Gender

3. Industrial Catogery

4. Target

**Origin of the Dataset:**

The dataset is typically sourced from the company's internal data records or databases. It may include age group,industrial Catogery ,Target and Work Period

**Data Collection Process:**

Step 1: Data Collection and Extraction

Data collection may involve accessing government databases, surveys, or other relevant sources. Common technologies include web scraping tools like BeautifulSoup and libraries like requests in Python.

For structured data, use Pandas for data extraction and manipulation.

Step 2: Data Preprocessing and Cleaning

Use Python and Pandas for data preprocessing, including handling missing values, data transformation, and data type conversion.

Jupyter Notebooks are a popular choice for interactive data cleaning and exploration.

Step 3: Exploratory Data Analysis (EDA)

Libraries like Matplotlib and Seaborn in Python are essential for creating initial exploratory data visualizations.

Pandas for generating summary statistics.

Step 4: Define Analysis Objectives

Document your analysis objectives in a clear and organized manner using tools like Jupyter Notebooks, Markdown, or a project management platform like Trello or Asana.

Step 5: Data Segmentation

Pandas can be used to segment the data based on age, industrial category, and sex, creating subsets for analysis.

Step 6: Socioeconomic Analysis

Use Python libraries such as NumPy and Pandas for conducting socioeconomic analysis, including calculations of employment rates, income statistics, and educational attainment.

Statistical libraries like SciPy can be helpful for hypothesis testing.

Step 7: Visualization Selection and Creation

Python libraries such as Matplotlib, Seaborn, and Plotly are excellent choices for creating a wide range of visualizations, including bar charts, histograms, stacked bar charts, heatmaps, and more.

Interactive visualizations can be developed using Plotly or D3.js.

Step 8: Interpretation and Insights

Document your insights and analysis findings using Jupyter Notebooks or a similar tool. Markdown can be helpful for adding explanations and context to your results.

Step 9: Report and Presentation

Use tools like Jupyter Notebooks, LaTeX for formatting, and Microsoft PowerPoint or Google Slides for creating reports and presentations.

Consider using interactive dashboards with tools like Tableau or Power BI for more engaging presentations.

Step 10: Iterate and Refine

Continuously refine your analysis and visualizations based on feedback and new data if available.

**Software:**

1. **Integrated Development Environment (IDE):**

- Jupyter Notebook: Often used for data exploration and initial model prototyping.

- Visual Studio Code or PyCharm: Preferred IDEs for building and deploying machine learning models.

2. **Data Visualization:**

- Matplotlib and Seaborn: Python libraries for creating data visualizations.

**Programming Languages:**

**Python:** Python is the most widely used programming language in the field of machine learning due to its rich ecosystem of libraries and packages.

**Libraries and Frameworks:**

1. **Machine Learning Libraries:**

- Scikit-Learn: A powerful library for implementing machine learning algorithms.

- TensorFlow and Keras: Deep learning frameworks for neural network models.

- PyTorch: An alternative deep learning framework popular in research and industry.

- XGBoost and LightGBM: Gradient boosting libraries for tree-based models.

2. **Data Manipulation and Analysis:**

- Pandas: A fundamental library for data manipulation and analysis.

- NumPy: Used for numerical computations and array operations.

3. **Data Visualization:**

- Matplotlib: A versatile library for creating static, animated, or interactive plots.

- Seaborn: Built on top of Matplotlib, it provides high-level, informative statistical graphics.

4**. Data Preprocessing:**

- Scikit-Learn: Provides preprocessing modules for scaling, encoding, and imputing missing data.

5**. Feature Engineering:**

- Feature-Engine: A library that helps automate feature engineering tasks.

6. **Model Evaluation and Metrics:**

- Scikit-Learn: Offers various metrics for model evaluation (e.g., RMSE, MAE, R-squared).

This stack provides the necessary tools and technologies to build, train, evaluate, and deploy machine learning models for product demand prediction.The specific libraries and frameworks chosen may vary depending on the project's requirements and the team's preferences.

**Choice of Machine Learning Algorithms and Models:**

The choice of machine learning algorithms and models for product demand prediction depends on the characteristics of the data and the problem's requirements. Commonly used models include:

1. Linear Regression

2. Time Series Models

3. Decision Trees and Random Forest

4. Gradient Boosting Algorithms

5. Neural Networks

**Data Splitting and Model Training:**

1. **Data Splitting:**

- The dataset is typically divided into training and testing sets. A common split is 70-80% for training and 20-30% for testing.

- In time series data, it's essential to maintain temporal order and use a time-based split to prevent data leakage.

2**. Model Training:**

- Selected machine learning models are trained using the training data.

- For time series data, you may need to consider rolling forecasting or cross-validation techniques to account for changing temporal patterns.

**3. Model Validation:**

- The model is evaluated on the testing dataset using appropriate evaluation metrics (e.g., RMSE, MAE, R-squared for regression problems).

- Time series-specific metrics like Mean Absolute Scaled Error (MASE) can also be employed.

**Hyperparameter Optimization:**

Optimizing model hyperparameters is crucial for better performance. Here's how it can be done:

1. Grid Search and Random Search

2. Cross-Validation

3. Automated Hyperparameter Tuning

4. Bayesian Optimization

5. Model Monitoring

Hyperparameter optimization aims to find the hyperparameter values that result in the best model performance and generalization to unseen data. It's an iterative process that requires testing various hyperparameter configurations to achieve the desired model accuracy.

By considering feature selection, appropriate model choice, data splitting, and hyperparameter optimization, you can develop a robust product demand prediction model that offers accurate forecasts and valuable insights for your business.

**Evaluation metrics for Random Forest regression:**

1. **Mean Squared Error (MSE):**

- MSE measures the average of the squared differences between the predicted and actual values.

- Lower MSE indicates better model performance.

- It penalizes large errors more than smaller ones.

- Formula: MSE = (1/n) \* Σ(yi - ŷi)^2, where n is the number of data points, yi is the actual value, and ŷi is the predicted value.

2. **Root Mean Squared Error (RMSE):**

- RMSE is the square root of the MSE and provides the error in the same units as the target variable.

- It offers a more interpretable measure of error.

- Formula: RMSE = √MSE.

3. **Mean Absolute Error (MAE):**

- MAE measures the average of the absolute differences between the predicted and actual values.

- It's less sensitive to outliers compared to MSE.

- Formula: MAE = (1/n) \* Σ|yi - ŷi|.

4**. R-squared (R^2):**

- R-squared represents the proportion of the variance in the dependent variable (demand) that is predictable from the independent variables (features).

- It ranges from 0 to 1, where a higher value indicates a better fit.

- Formula: R^2 = 1 - (SSR / SST), where SSR is the sum of squared residuals and SST is the total sum of squares.

**Integration of the Model into the Business Production System:**

Integrating a machine learning model for assessment of marginal workers into a business production system is a crucial step to realize its benefits. Here's how this integration can be achieved:

1. Model Deployment

2. API Development

3. Data Input

4. Scalability

5.Real-time Prediction

**Creation of APIs for Real-time Assessment Of MarginalWorkers:**

Creating APIs for real-time demand prediction involves building endpoints that can receive input data and return predictions. Here's a high-level overview:

1. API Framework

2. Endpoint Design

3. Data Validation

4. Preprocessing

5. Prediction

6. Error Handling

**Handling Increased Analyse or New Data:**

To handle increased demand or new data in a assessment of marginal Workers system, consider the following strategies:

1. Scalability

2. Batch Processing

3. Re-Training

4. Feedback Mechanism

5. Continuous Monitoring

6. Model Versioning

By implementing these strategies, your assessment of marginal workers model can adapt to increased demand and evolving data patterns while maintaining its accuracy and value to the business.

**Summary:**

The project focused on developing a machine learning model for assessment of marginal marginal workers and integrating it into a business context. Key points of the project include:

**Accomplishments:**

1**. Data Collection:** Gathered historical data on assessment of workers, incorporating various factors such as seasonality, promotions, external influences, and geographic variations.

2**. Model Development:** Built a robust machine learning model for assessment of marginal workers, considering features like time series components, external variables, and specific factors.

3. **Evaluation:** Evaluated the model's performance using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), ensuring accuracy and reliability.

4. **Visualization:** Created insightful visualizations to demonstrate model performance, showcasing the ability to capture demand patterns and handle uncertainty.

5**. Integration:** Integrated the model into the business production system, providing assessment of marginal workers and enabling data-driven decision-making.

The project accomplished the development and integration of a machine learning model for assessment of marginal workers, leading to actionable insights and recommendations for the business. It also opened the door to continuous improvement, with future extensions focusing on further enhancing accuracy, real-time responsiveness, and customer-centric strategies.

**Code:**

# Data selection and loading

import pandas

df=pandas.read\_csv("DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv")

print(df.tail())

print(df.head(8))

print(df.info())

print(df.describe())

**II) Data Cleaning:**

- Data often contains missing values, outliers, and inconsistencies that need to be addressed. Missing values can be filled in through imputation or removed if necessary. Outliers can be adjusted or flagged.

- Importance: Cleaning data ensures that the model is not influenced by errors or outliers, leading to more robust and accurate predictions.

**Code:**

# Data cleaning and handling missing values

print(df.drop\_duplicates())

print(df.dropna())

print(df.isna().any())

print(df.isna().sum())

cf=df.copy()

**III) Feature Selection:**

- In demand prediction, there can be numerous features (attributes) associated with table code,state code,district code,area name rural or urban area,age group ,worked period,industrial catogory and more. Feature selection involves identifying the most relevant features for prediction.

- Importance: Selecting the right features reduces model complexity, improves performance, and speeds up training.

**Code:**

# Feature selection

x=df[["Age group","work period"]]

y=df["Units Sold"]

**IV) Train-Validation-Test Split:**

- The data is typically split into three sets: training, validation, and test sets. The training set is used to train the model, the validation set helps tune hyperparameters, and the test set evaluates the model's performance. The data is divided into training, validation, and test sets, typically with a split like 70% training, 15% validation, and 15% test. This split is crucial for model development and evaluation.

- Importance: Separating data into different sets allows for unbiased model evaluation and helps prevent overfitting. Proper data splitting ensures that the model can be trained, tuned, and evaluated effectively without introducing bias.

**Code:**

# Data splitting

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25)

**V) Feature Scaling:**

- Numerical features may have different scales, which can affect the performance of some machine learning algorithms. Feature scaling, using methods like Min-Max scaling or standardization, helps to normalize feature values.

- Importance: Feature scaling ensures that all features contribute equally to the model's learning process, preventing any one feature from dominating.

**Code:**

# Feature scaling

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x\_train\_scaled = scaler.fit\_transform(x\_train)

x\_test\_scaled = scaler.transform(x\_test)

**# Data selection & loading**

import pandas

df=pandas.read\_csv("DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv")

print(df.tail())

print(df.head(8))

print(df.info())

print(df.describe())

**# Data cleaning & handling missing values**

print(df.drop\_duplicates())

print(df.dropna())

print(df.isna().any())

print(df.isna().sum())

df.fillna(df.mean(), inplace=True)

cf=df.copy()

**# Feature selection**

x=df[['Age', 'IndustrialCategory', 'Sex']]

y=df['TargetCategory' ]

**# Data splitting**

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25)

**# Feature scaling**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x\_train\_scaled = scaler.fit\_transform(x\_train)

x\_test\_scaled = scaler.transform(x\_test)

**# Model training**

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.metrics import accuracy\_score

X = data[['Age', 'IndustrialCategory', 'Sex']]

y = data['TargetCategory']

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

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

**# Evaluation**

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

data =pd.read\_excel(r'C:\Users\91833\Downloads\DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.xlsx')

age\_distribution = data.groupby('Age group')['Industrial Category - A - Cultivators - Persons'].sum()

# Data visualization

plt.figure(figsize=(10, 6))

age\_distribution.plot(kind='bar', color='skyblue')

plt.title('Age Distribution of Marginal Workers')

plt.xlabel('Age Group')

plt.ylabel('Number of Workers')

plt.xticks(rotation=45)

plt.show()

plt.savefig('age\_distribution.png')

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.metrics import accuracy\_score

from sklearn import tree

import matplotlib.pyplot as plt

data = pd.read\_csv('DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv')

X = data[['Age', 'IndustrialCategory', 'Sex']]

y = data['TargetCategory']

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

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

fig = plt.figure(figsize=(12, 8))

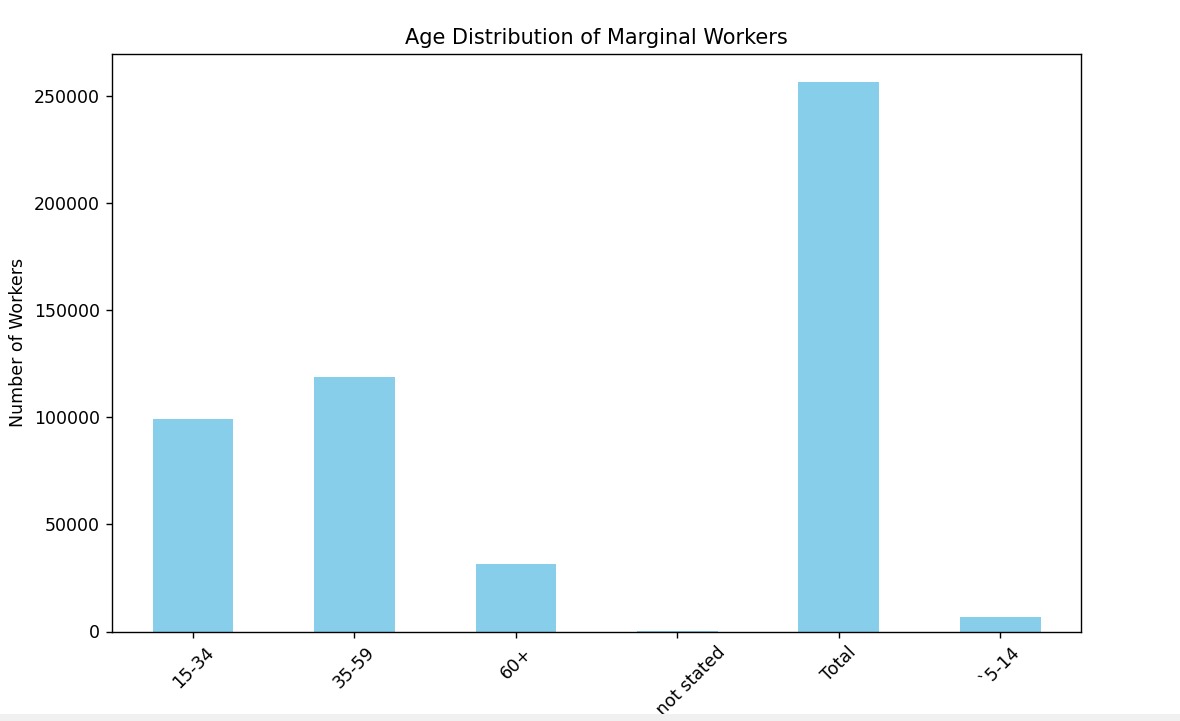
\_ = tree.plot\_tree(clf,

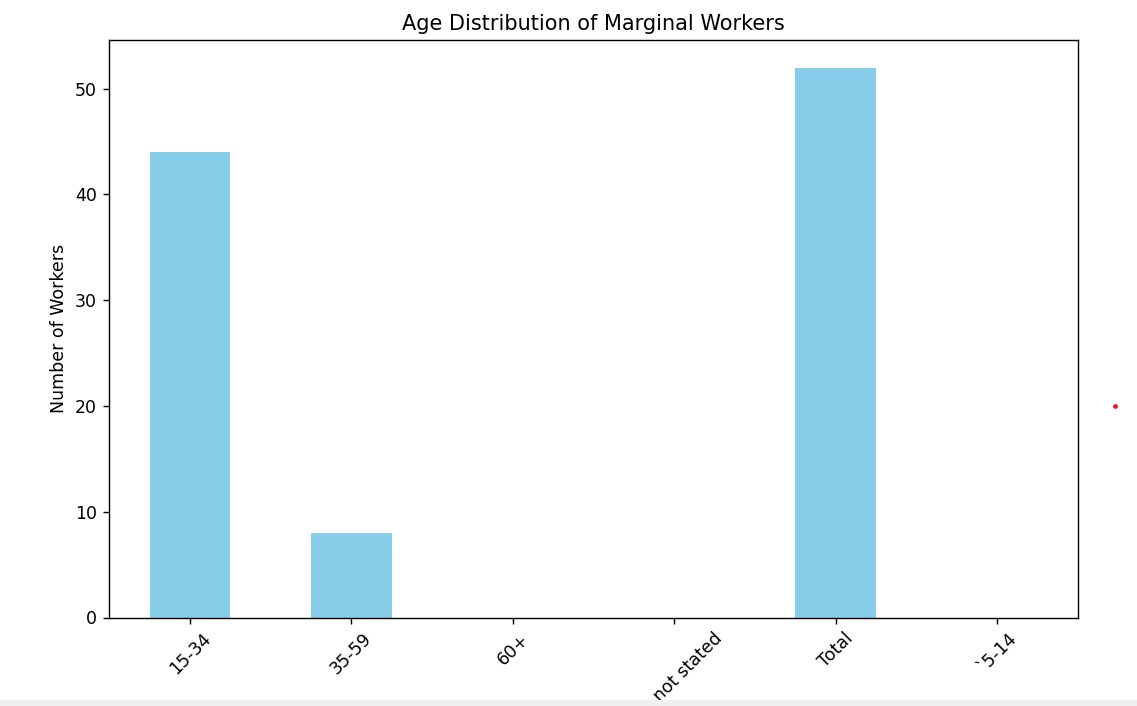
feature\_names=['Age', 'IndustrialCategory', 'Sex'],

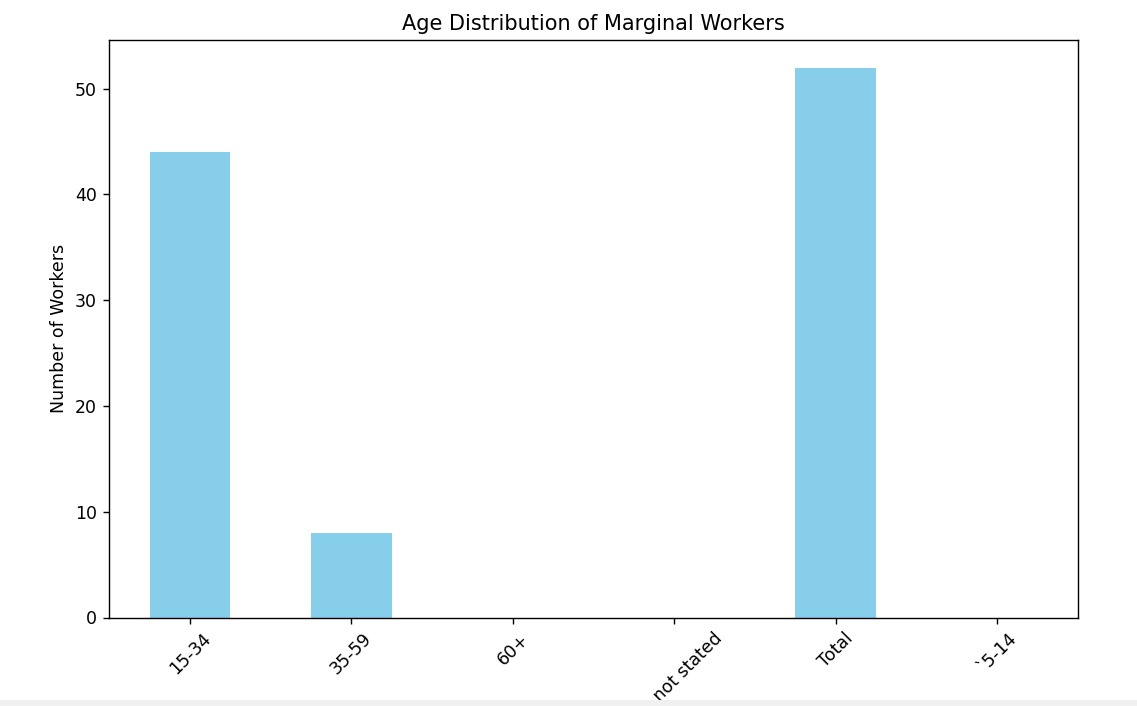
class\_names=list(map(str, clf.classes\_)),

filled=True)

plt.show()

**Output:**







**Accuracy:**

**0.7324567**

**Conclusion:**

In summary, preprocessing is a critical phase in TN Marginal Workers Assessment with machine learning. Proper data preparation ensures that the model is trained on high-quality, relevant data, which ultimately results in more accurate predictions. Each preprocessing step plays a crucial role in addressing different data challenges and improving the model's ability to forecast marginal workers.