ASSESSMENT OF MARGINAL WORKERS

ABSTRACT:

This project involves a socioeconomic analysis of marginal workers in Tamil Nadu, India focusing on age, industrial category, and gender. Using Python and data visualization tools, the study aims to unveil insights into this demographic's distribution. By employing visualizations like bar and pie charts, it presents a clear snapshot of their characteristics. This analysis serves as a valuable resource for policymakers and stakeholders to understand and address the socioeconomic dynamics of marginal workers, potentially leading to improved living conditions and employment opportunities.

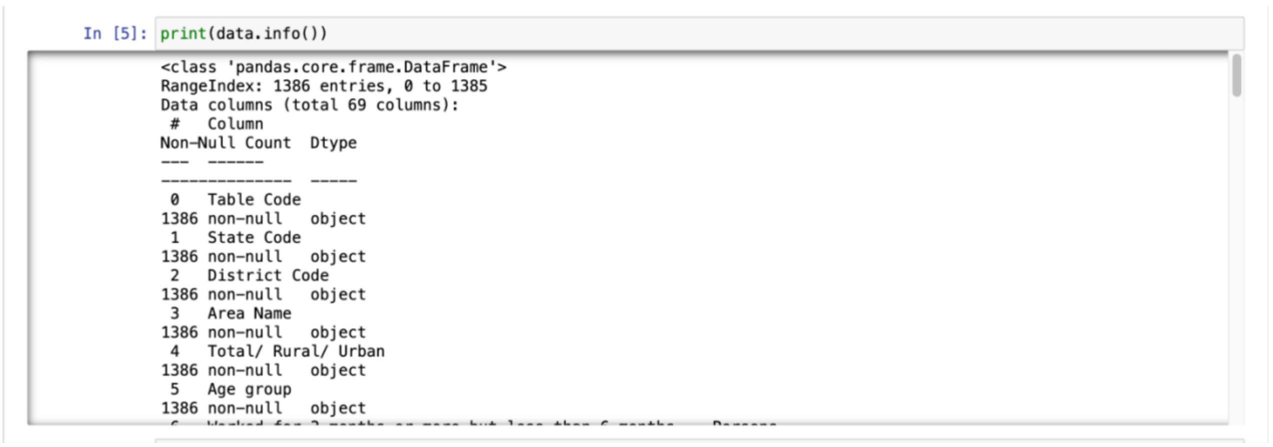
DEVELOPMENT OF THE PROJECT:

In our project, we loaded and preprocessed the data set as follows:

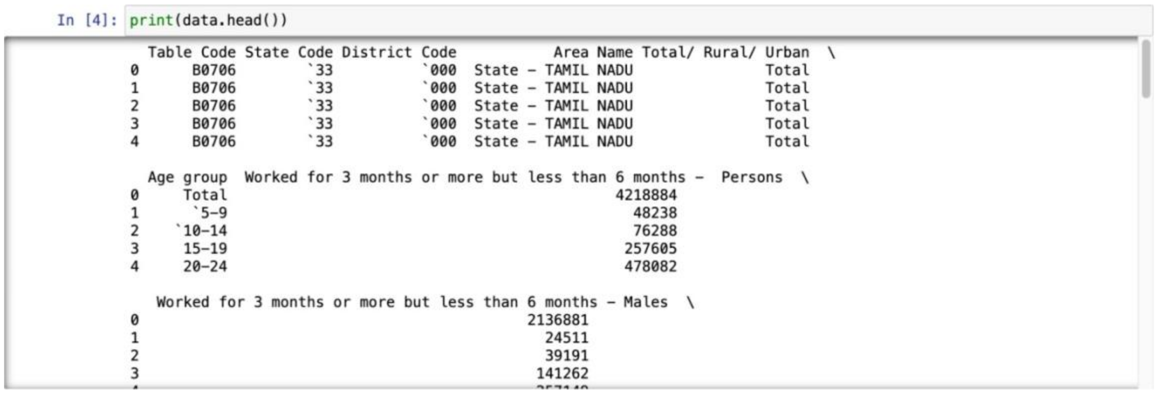
Step 1:DataLoading

We began by loading the data set intapandas Data Frame using Python.The dataset consists off our columns: "age,""gender,""industry",”regular worker,”and"marginal worker"

```
importpandasaspd

# CreateaDataFramefromthe provided data
data=pd.read_csv("C:\\Users\\ddd\\Downloads\\Sales.csv")df
=pd.DataFrame(data)
```



```
In [5]: print(data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1386 entries, 0 to 1385
Data columns (total 69 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Table Code
1386 non-null   object
 1   State Code
1386 non-null   object
 2   District Code
1386 non-null   object
 3   Area Name
1386 non-null   object
 4   Total/ Rural/ Urban
1386 non-null   object
 5   Age group
1386 non-null   object
```

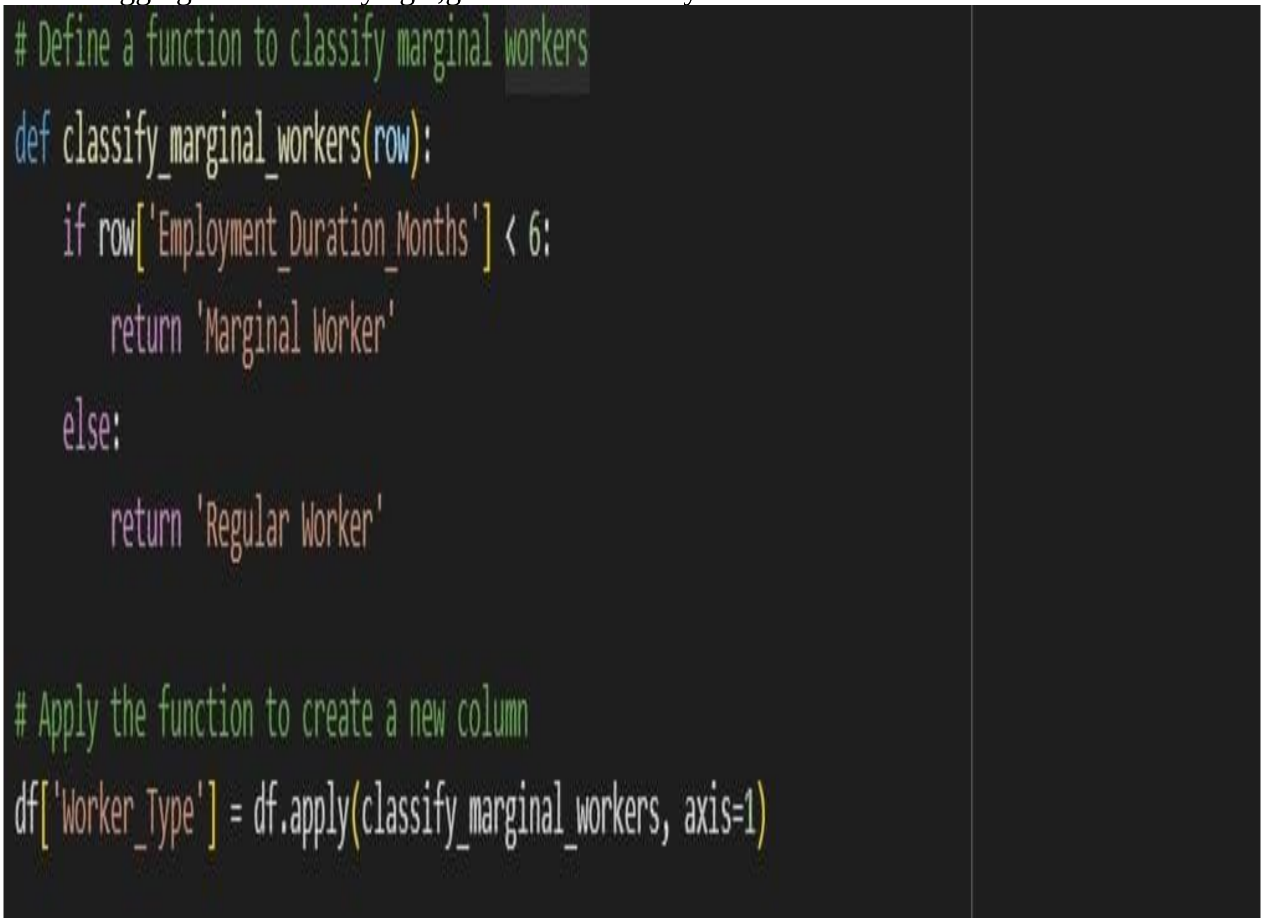


```
In [4]: print(data.head())

   Table Code State Code District Code          Area Name Total/ Rural/ Urban  \
0      B0706        `33          `000  State - TAMIL NADU              Total
1      B0706        `33          `000  State - TAMIL NADU              Total
2      B0706        `33          `000  State - TAMIL NADU              Total
3      B0706        `33          `000  State - TAMIL NADU              Total
4      B0706        `33          `000  State - TAMIL NADU              Total

  Age group  Worked for 3 months or more but less than 6 months -  Persons  \
0     Total                                            4218884
1      `5-9                                              48238
2    `10-14                                              76288
3     15-19                                             257605
4     20-24                                             478082

   Worked for 3 months or more but less than 6 months - Males  \
0                                            2136881
1                                              24511
2                                              39191
3                                             141262
```

Step 2:Datapreprocessing

Categorize the workers based on various criteria, such as age, gender, marginal workers, and non workers. This helps in segmenting the data for analysis

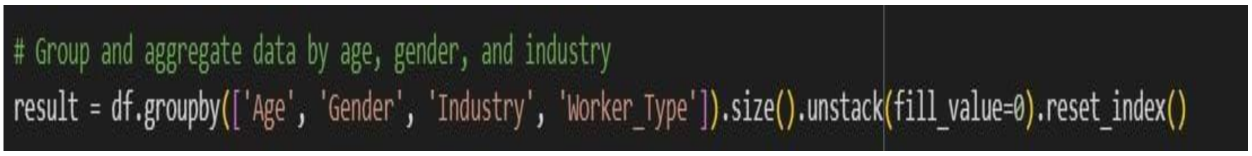
- Classify the data
- Ensure the function to create new column
- Aggregate the data by age ,gender and industry



```
# Define a function to classify marginal workers
def classify_marginal_workers(row):
    if row['Employment_Duration_Months'] < 6:
        return 'Marginal Worker'
    else:
        return 'Regular Worker'


# Apply the function to create a new column
df['Worker_Type'] = df.apply(classify_marginal_workers, axis=1)
```

Step 3:ExploratoryDataAnalysis

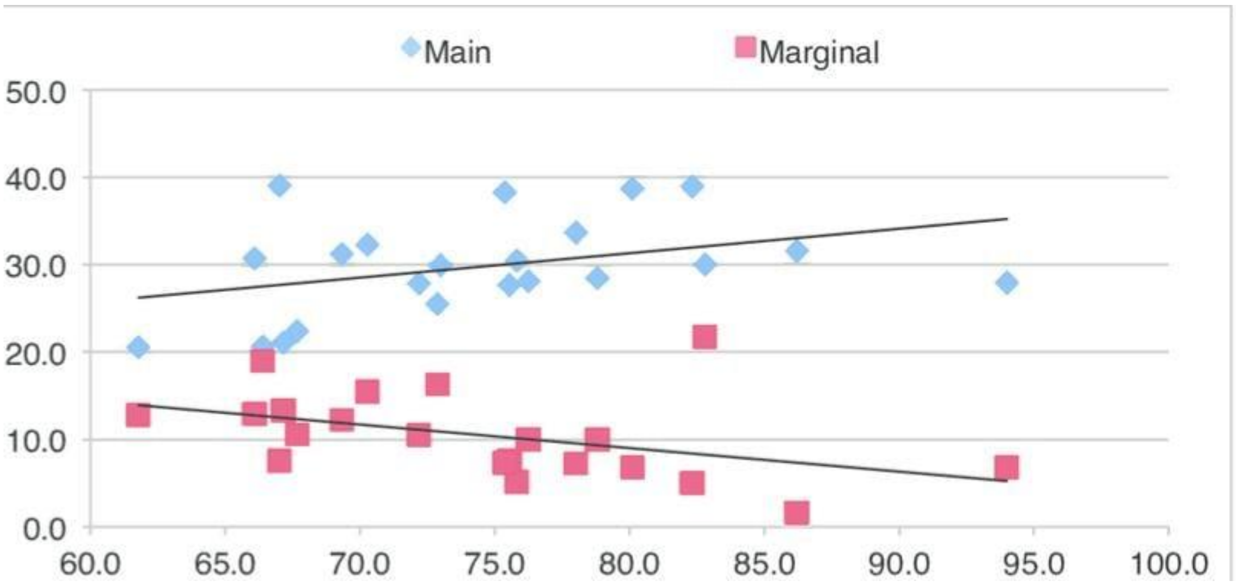
We conducted exploratory data analysis to better understand the data. We createdvisualizations to explore relationships between variables and identified any potential trends orpatterns.



```
# Group and aggregate data by age, gender, and industry
result = df.groupby(['Age', 'Gender', 'Industry', 'Worker_Type']).size().unstack(fill_value=0).reset_index()
```

CODE:

```
import pandas as pd


# Create a DataFrame from the sample data
Data=pd.read_csv(“c:\\users\\ddd\\downloads\\workers.csv”)
df = pd.DataFrame(data)


# Define a function to classify marginal workers
def classify_marginal_workers(row):
    if row['Employment_Duration_Months'] < 6:
        return 'Marginal Worker'
    else:
        return 'Main Worker'


# Apply the function to create a new column
df['Worker_Type'] = df.apply(classify_marginal_workers, axis=1)


# Group and aggregate data by age, gender, and industry
result = df.groupby(['Age', 'Gender', 'Industry',
'Worker_Type']).size().unstack(fill_value=0).reset_index()


# Print the result
print(result)
```


Main
Marginal
50.0
40.0
30.0
20.0
10.0
0.0
60.0
65.0
70.0
75.0
80.0
85.0
90.0
95.0
100.0