Assessment of Marginal Workers In Tamilnadu

PHASE-5

Analysis Approach:

The analysis approach outlines the methodology and techniques you'll use to achieve the project's objectives. This can include data collection, preprocessing, modeling, and evaluation strategies. For example:

Data Collection: We collected customer data, including demographics, subscription details, and usage patterns from the company's database.

Data Preprocessing: Cleaned the data, handled missing values, and encoded categorical variables.

Modeling: Employed logistic regression, decision trees, and random forests to predict customer churn.

Evaluation: Used metrics such as accuracy, precision, recall, and ROC-AUC to assess model performance.

Visualization Types:

Visualization is a crucial part of data analysis to communicate insights effectively. The types of visualizations will depend on the data and the objectives. For example:

Bar charts and pie charts to show customer demographics and subscription breakdown.

Line plots to visualize stock price trends over time.

Word clouds to display frequently mentioned terms in social media sentiment analysis.

Code Implementation:

This section provides a high-level overview of the code and tools used in the project. For example:

We implemented the analysis using Python and popular libraries like Pandas, NumPy, and Scikit-Learn.

Data preprocessing was carried out using Pandas DataFrame manipulation.

We created visualizations using Matplotlib and Seaborn.

Machine learning models were built using Scikit-Learn and XGBoost.

It's important to note that the specifics of project objectives, analysis, visualization, and code implementation will vary greatly from one project to another. The details and complexity of each aspect will depend on the project's domain, the available data, and the tools and techniques chosen for analysis.

About the project

Following the given instructions we build our Project

They are:

Perform the demographic analysis
Calculate the distribution of marginal workers based on age, industrial category, and se using data aggregation and manipulation
Create visualizations

Create visualizations using data visualization libraries (e.g., Matplotlib, Seaborn).

Downloading datasets

The dataset has been download for the given link

The dataset link :

https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states

Then build and preprocessing the data set

Import the files

```
#import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#loading dataset
df= pd.read_csv(“marginal.csv”)
df.head()
```

Output

Python

Area Name	Total/ Rural/ Urban	Age group	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	Worked for less than 3 months - Persons	...	Industrial Category - N to O - Females	Industrial Category - P to Q - Persons	Industrial Category - P to Q - Males	Industrial Category - P to Q - Females	Industrial Category - R to U - HHI - Persons	Industrial Category - R to U - HHI - Males	Industrial Category - R to U - HHI - Females	Industrial Category - R to U - Non HHI - Persons	Industrial Category - R to U - Non HHI - Males	Industrial Category - R to U - Non HHI - Females
State - TAMIL NADU	Total	Total	1200828	589003	611825	221386	...	3565	11080	4019	7061	16833	4266	12567	122088	55801	66287
State - TAMIL NADU	Total	`5-14	27791	14125	13666	2447	...	11	122	71	51	427	169	258	19305	9774	9531
State - TAMIL NADU	Total	15-34	514340	259560	254780	92423	...	1754	7536	2718	4818	8346	2127	6219	68929	32803	36126
State - TAMIL NADU	Total	35-59	542581	251957	290624	99202	...	1619	3205	1131	2074	6591	1487	5104	26498	9675	16823

Classified the data in sex & age

```
data = {
Age: [20, 30, 40, None 30, 40, 20, 30, 40). # Adding a missing value
Industrial Category: [‘N to 0- Females’, ‘P to Q – Fersons’ “P to Q Females, R to UHHI Pe R to UHHI – Females. R
to U-No
Sex: [‘Male, Female Male, Female Male’, ‘Female’. Count 100, 200, 120, 80, 150, 60, 90, 130, 50]

}
df pd DataFrame(data)
#Handling missing values

Df=df dropna()
#Print the cleaned data

print(df)
```

Output:

	Age	Industrial_Category	Sex	Count
0	20.0	N to O - Females	Male	100
1	30.0	P to Q - Persons	Female	200
2	40.0	P to Q - Males	Male	120
4	30.0	R to U - HHI - Persons	Male	150
5	40.0	R to U - HHI - Males	Female	60
6	20.0	R to U - HHI - Females	Male	90
7	30.0	R to U - Non HHI - Persons	Female	130
8	40.0	R to U - Non HHI - Males	Male	50

Data aggregation

#Data aggregation
agg_data=df groupby([Age Industrial Category", "Sex"]).size()
#Print the aggregate data

Print(agg_data)

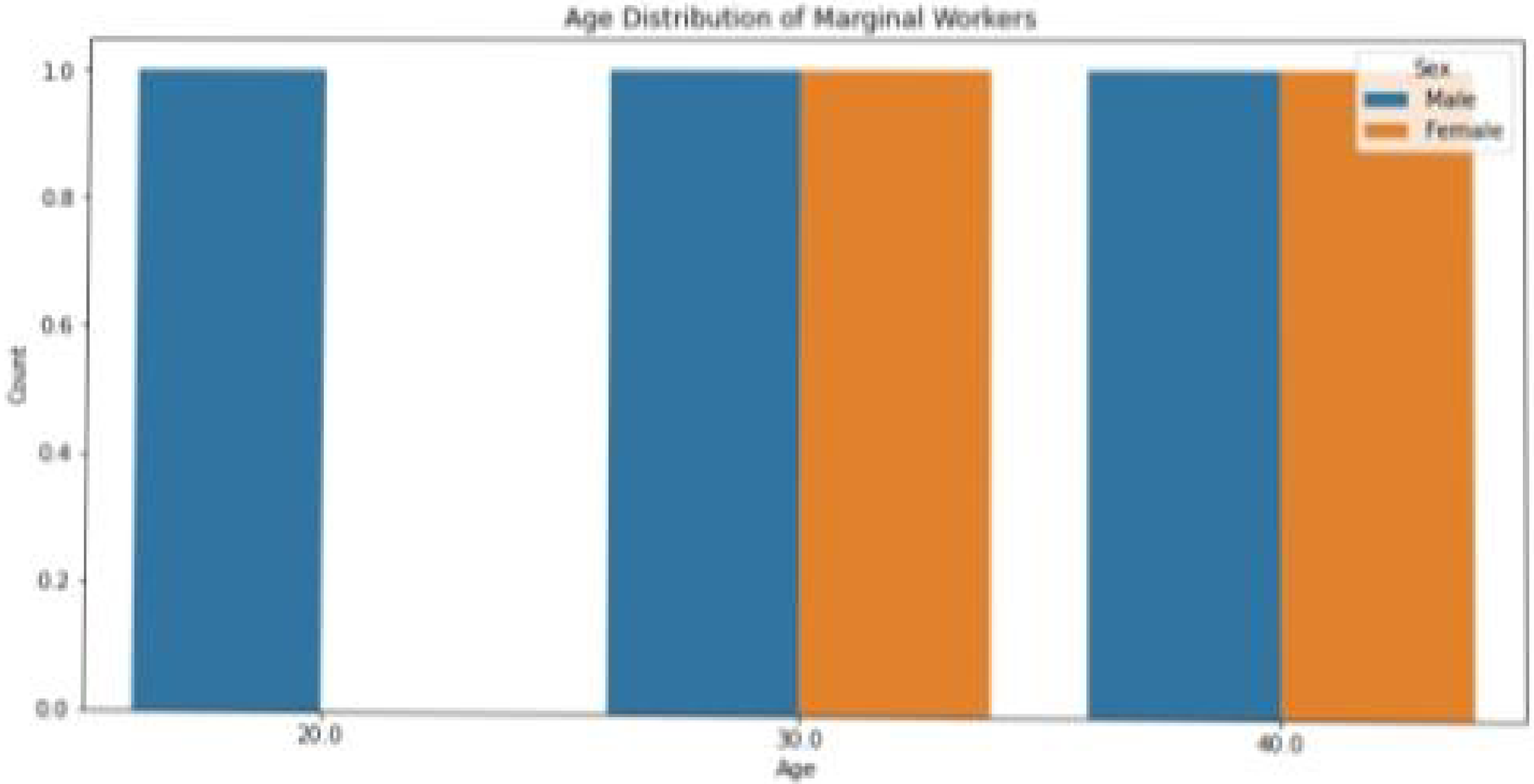
Output:

	Age	Industrial_Category	Sex	Count
0	20.0	N to O - Females	Male	1
1	20.0	R to U - HHI - Females	Male	1
2	30.0	P to Q - Persons	Female	1
3	30.0	R to U - HHI - Persons	Male	1
4	30.0	R to U - Non HHI - Persons	Female	1
5	40.0	P to Q - Males	Male	1
6	40.0	R to U - HHI - Males	Female	1
7	40.0	R to U - Non HHI - Males	Male	1

Age distribution plotting point

```
#Plot Age Distribution
plt figure( figsize=(12, 6))
sns barplot(x=Age,y=Count, data= agg_data, hue ‘Sex’)
plt.title (‘Age Distribution of Marginal Workers’)
plt.xlabel(‘Age’)

plt.ylabel (‘Count’)
plt.legend (title=‘Sex’,loc=‘upper right’)
plt show()
```

Output

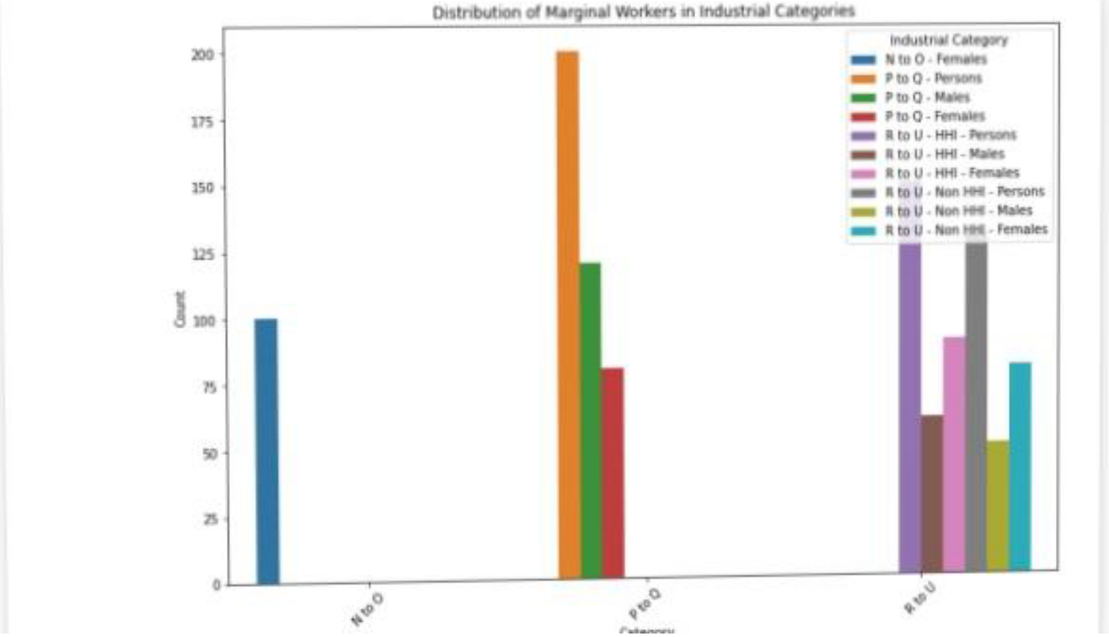

Age Distribution of Marginal Workers
Count
1.0
0.8
0.6
0.4
0.2
0.0
20.0
30.0
40.0
Age
Sex
Male
Female

Data category

In [9]:

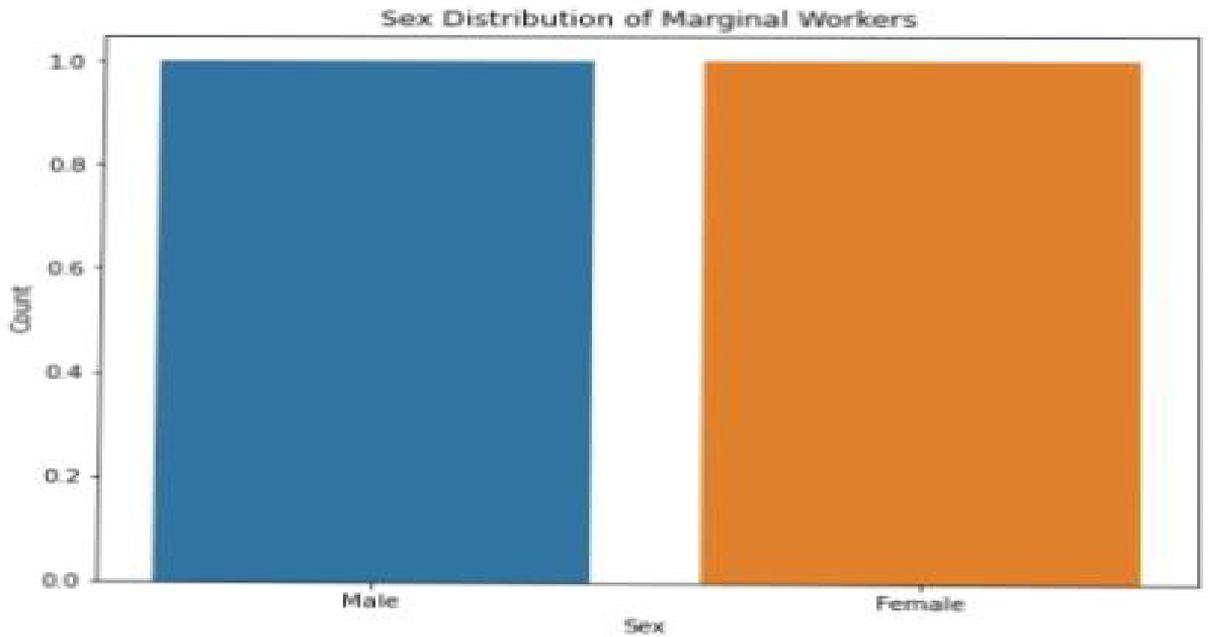
```
data = {
    'Industrial Category': ['N to O - Females', 'P to Q - Persons'
                            'R to U - HHI - Persons', 'R to U - HH
                            'R to U - Non HHI - Persons', 'R to U
    'Count': [100, 200, 120, 80, 150, 60, 90, 130, 50, 80]
}

df = pd.DataFrame(data)

# Extracting category information for better plotting
df['Category'] = df['Industrial Category'].str.split('-').str[0].s

# Plotting
plt.figure(figsize=(12, 8))
sns.barplot(x='Category', y='Count', hue='Industrial Category', da
plt.title('Distribution of Marginal Workers in Industrial Categori
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readabi
plt.show()
```

Output


Distribution of Marginal Workers in Industrial Categories
Industrial Category
N to O - Females
P to Q - Persons
P to Q - Males
P to Q - Females
R to U - HHI - Persons
R to U - HHI - Males
R to U - HHI - Females
R to U - Non HHI - Persons
R to U - Non HHI - Males
R to U - Non HHI - Females
Count
0
25
50
75
100
125
150
175
200
N to O
P to Q
R to U

Sex distribution plot

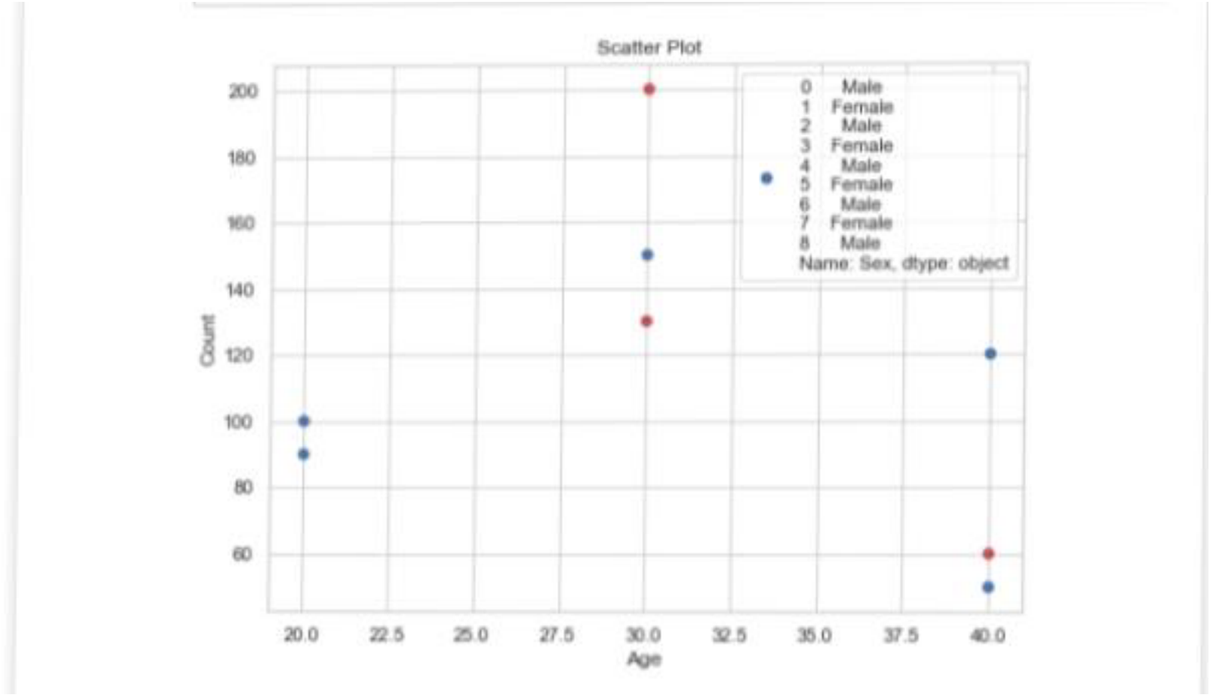
```
plt.figure(figsize=(8, 6))
sns.barplot(x='Sex', y='Count', data=agg_data)
plt.title('Sex Distribution of Marginal Workers')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```



Scatterplot

In [13]:
```
# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Count'], c=df['Sex'].map({'Male': 'b',
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```

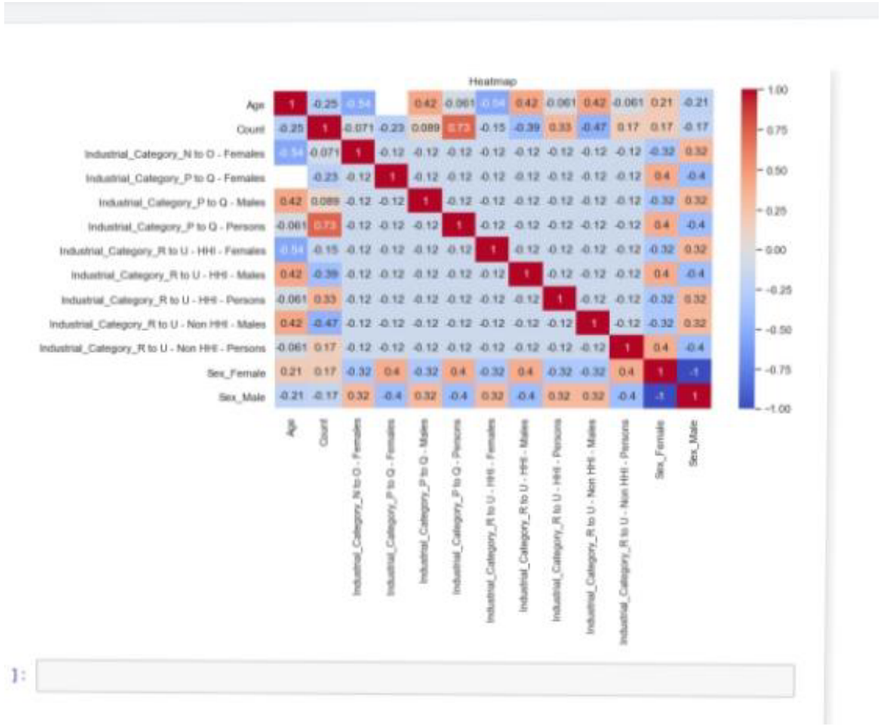
Output


Scatter Plot
0 Male
1 Female
2 Male
3 Female
4 Male
5 Female
6 Male
7 Female
8 Male
Name: Sex, dtype: object
Count
200
180
160
140
120
100
80
60
20.0
22.5
25.0
27.5
30.0
32.5
35.0
37.5
40.0
Age

Heatmap

```
#Create a heatmap

 plt.figure (figsize=(10,6))
sns.heatmap(df_encode.Corr(),Annot=true,Camp='coolwarm')
plt.tittle('heatmap')

Plt.show()
```

Output


Heatmap
Age
Count
Industrial_Category_N to O - Females
Industrial_Category_P to Q - Females
Industrial_Category_P to Q - Males
Industrial_Category_P to Q - Persons
Industrial_Category_R to U - HHI - Females
Industrial_Category_R to U - HHI - Males
Industrial_Category_R to U - HHI - Persons
Industrial_Category_R to U - Non HHI - Males
Industrial_Category_R to U - Non HHI - Persons
Sex_Female
Sex_Male
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Finding missing value

#find missing values in the dataset
df.isnull().

sum(axis=0)

```
Table Code                                        0
State Code                                        0
District Code                                     0
Area Name                                         0
Total/ Rural/ Urban                               0
                                                 ..
Industrial Category - R to U - HHI - Males        0
Industrial Category - R to U - HHI - Females      0
Industrial Category - R to U - Non HHI - Persons  0
Industrial Category - R to U - Non HHI - Males    0
Industrial Category - R to U - Non HHI - Females  0
Length: 69, dtype: int64
```

Retrieve and Testing the dataset

```
# Retrieve training and testing dataset
X = df.iloc[:,:-1]
Y = df.iloc[:,-1
print(X)
```

Output

```
     Table Code State Code District Code             Area Name  \
0       B0806SC         `33           `000   State - TAMIL NADU
1       B0806SC         `33           `000   State - TAMIL NADU
2       B0806SC         `33           `000   State - TAMIL NADU
3       B0806SC         `33           `000   State - TAMIL NADU
4       B0806SC         `33           `000   State - TAMIL NADU
..          ...         ...            ...                  ...
589     B0806SC         `33           `633  District - Tiruppur
590     B0806SC         `33           `633  District - Tiruppur
591     B0806SC         `33           `633  District - Tiruppur
592     B0806SC         `33           `633  District - Tiruppur
593     B0806SC         `33           `633  District - Tiruppur

     Total/ Rural/ Urban       Age group  \
0                  Total           Total
1                  Total           `5-14
2                  Total           15-34
3                  Total           35-59
4                  Total             60+
..                   ...             ...
589                Urban           `5-14
590                Urban           15-34
591                Urban           35-59
592                Urban             60+
593                Urban  Age not stated
...
592                                      35
593                                       0
```

```
X = df.iloc[:,:-1]
Y = df.iloc[:,-1]
print(Y
```

```
0      66287
1       9531
2      36126
3      16823
4       3671
       ...
589      124
590      428
591      176
592       46
593        0
Name: Industrial Category - R to U - Non HHI - Females, Length: 594, dtype: int64
```

Select relevant Columns For analysis

```
# Select relevant columns for analysis
selected_columns = [‘Age group’, ‘Industrial Category – A – Cultivators – Persons’,
                    ‘Industrial Category – B – Persons’, ‘Industrial Category – C – HHI – Persons’,
                    ‘Industrial Category – D & E – Persons’, ‘Industrial Category – F – Persons’,
                    ‘Industrial Category – G – HHI – Persons’, ‘Industrial Category – H – Persons’,
                    ‘Industrial Category – I – Persons’, ‘Industrial Category – J – HHI – Persons’,
                    ‘Industrial Category – K to M – Persons’, ‘Industrial Category – N to O – Persons’,
                    ‘Industrial Category – P to Q – Persons’, ‘Industrial Category – R to U – HHI – Persons’]
```

Filter the dataframe

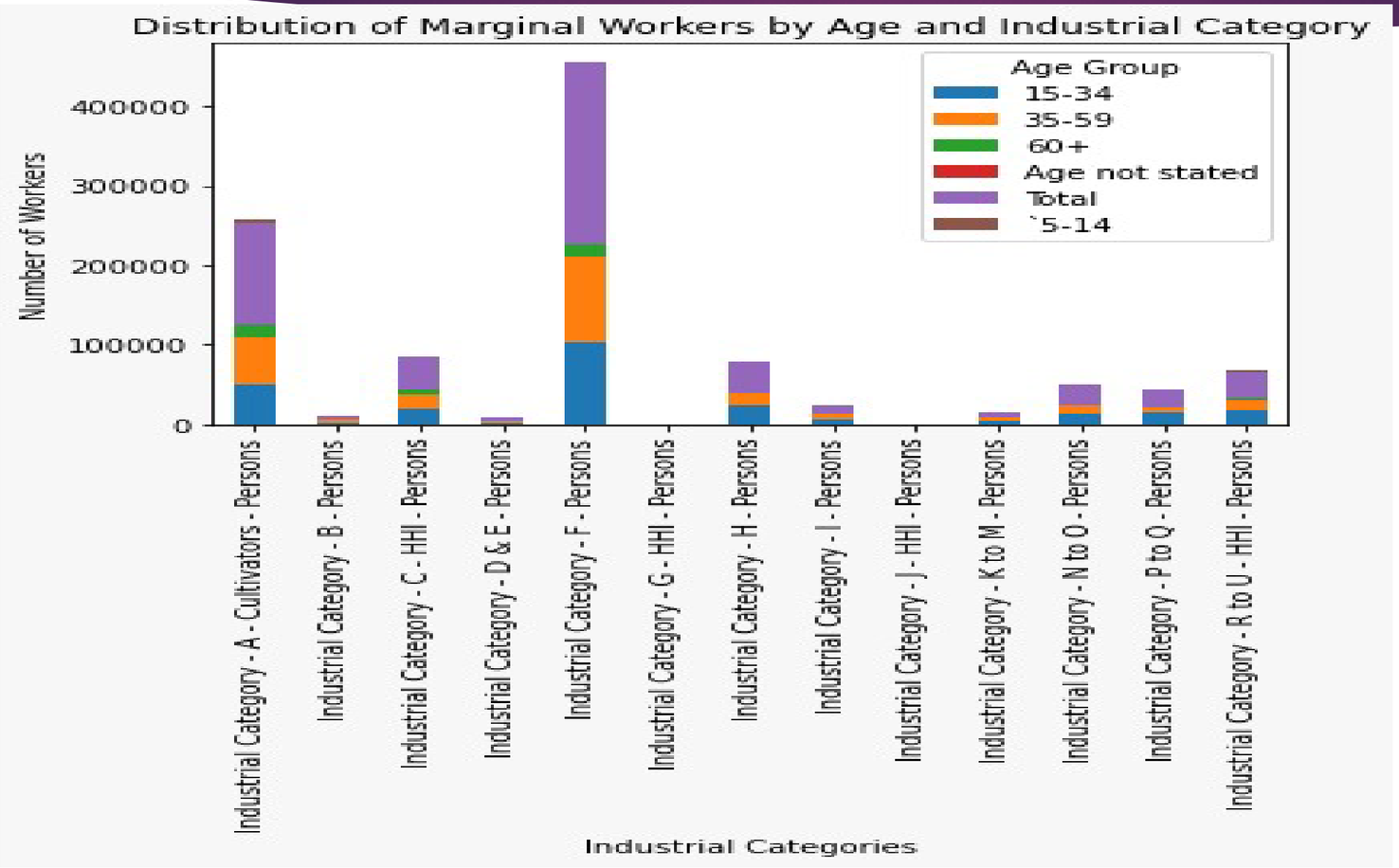
(‘# Filter the DataFrame for marginal workers and selected columns
marginal_workers_df = df[df[‘Total/ Rural/ Urban’] == ‘Total’][selected_columns]

# Group by age group and sum the counts for each industrial category
age_group_data = marginal_workers_df.groupbyAge group’).sum()

Plotting points

```
# Plotting
plt.figure(figsize=(15, 8))
age_group_data.T.plot(kind='bar', stacked=True)
plt.title('Distribution of Marginal Workers by Age and Industrial Category')
plt.xlabel('Industrial Categories')
plt.ylabel('Number of Workers')
plt.legend(title='Age Group')
plt.show()
```

Output:


Distribution of Marginal Workers by Age and Industrial Category
Age Group
15-34
35-59
60+
Age not stated
Total
`5-14
Number of Workers
0
100000
200000
300000
400000
Industrial Category - A - Cultivators - Persons
Industrial Category - B - Persons
Industrial Category - C - HHI - Persons
Industrial Category - D & E - Persons
Industrial Category - F - Persons
Industrial Category - G - HHI - Persons
Industrial Category - H - Persons
Industrial Category - I - Persons
Industrial Category - J - HHI - Persons
Industrial Category - K to M - Persons
Industrial Category - N to O - Persons
Industrial Category - P to Q - Persons
Industrial Category - R to U - HHI - Persons
Industrial Categories

Conclusion

The given dataset has been successfully analysing and Preprocessing the data