

TN MARGINAL WORKERS

Development part 2:

Introduction:

The Optimizing project success through Best practices for perform the demographic analysis and create visualization marginal worker Datasets in Tamil Nadu. Using matplotlib and libraries eg:seaborn

- ❖ Creating a visualization of marginal workers typically involves presenting data through charts or graphs. Various types of visualizations can be used depending on the specific information you want to convey. For example, you can consider creating pie charts to represent the distribution of different demographic groups, bar graphs to show the comparison of different parameters, or scatter plots to demonstrate relationships between variables.
- ❖ Data visualization is the graphical representation of data and information. It is commonly used to help people understand complex data through visual elements such as charts, graphs, and maps. Different types of data visualizations include bar charts, line graphs, pie charts, scatter plots, heat maps, and more. They are instrumental in spotting trends, correlations, and patterns within data, making it easier for people to interpret and comprehend information at a glance. If you have a specific dataset in mind, I can guide you on how to visualize it effectively.
- ❖ Matplotlib is a widely used Python library for creating static, interactive, and animated visualizations in Python. It is highly customizable and supports various types of plots such as line plots, scatter plots, bar plots, histograms, and many more. Here is a simple example to help you get started with a basic line plot using Matplotlib:

```
import matplotlib.pyplot as plt
```

```
# Sample data
```

```
x_values = [1, 2, 3, 4, 5]
```

```
y_values = [2, 4, 6, 8, 10]
```

```
# Plotting the data
```

```
plt.plot(x_values, y_values, marker='o')

# Adding title and labels
plt.title('Simple Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

# Display the plot
plt.show()
```

- ❖ Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Here's a simple example of a seaborn plot to get you started:

```
import seaborn as sns

import matplotlib.pyplot as plt

# Load example dataset
tips = sns.load_dataset("tips")

# Create a bar plot
sns.barplot(x="day", y="total_bill", data=tips)

# Show the plot
plt.show()
```

```
import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

from sklearn.metrics import silhouette_score

# Transforming the preprocessed data by extracting features and StandardScaler for clustering
df=pd.read_csv("preprocessed_data.csv")

columns=df.columns[6:48]

df1=pd.DataFrame(df,columns=columns)

scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(df1)

n_clusters = 2


# Create a K-Means clustering model
kmeans = KMeans(n_clusters=n_clusters, random_state=0)


# Fit the model to your scaled data
model=kmeans.fit(X_scaled)


# Get the cluster labels for each data point
labels = kmeans.labels_


# Get the coordinates of the cluster centers
cluster_centers = scaler.inverse_transform(kmeans.cluster_centers_)


# Show assignments for the first 3 data points
cluster_assignments = kmeans.labels_[:3]


# Cluster sizes
cluster_sizes = [sum(kmeans.labels_ == i) for i in range(n_clusters)]

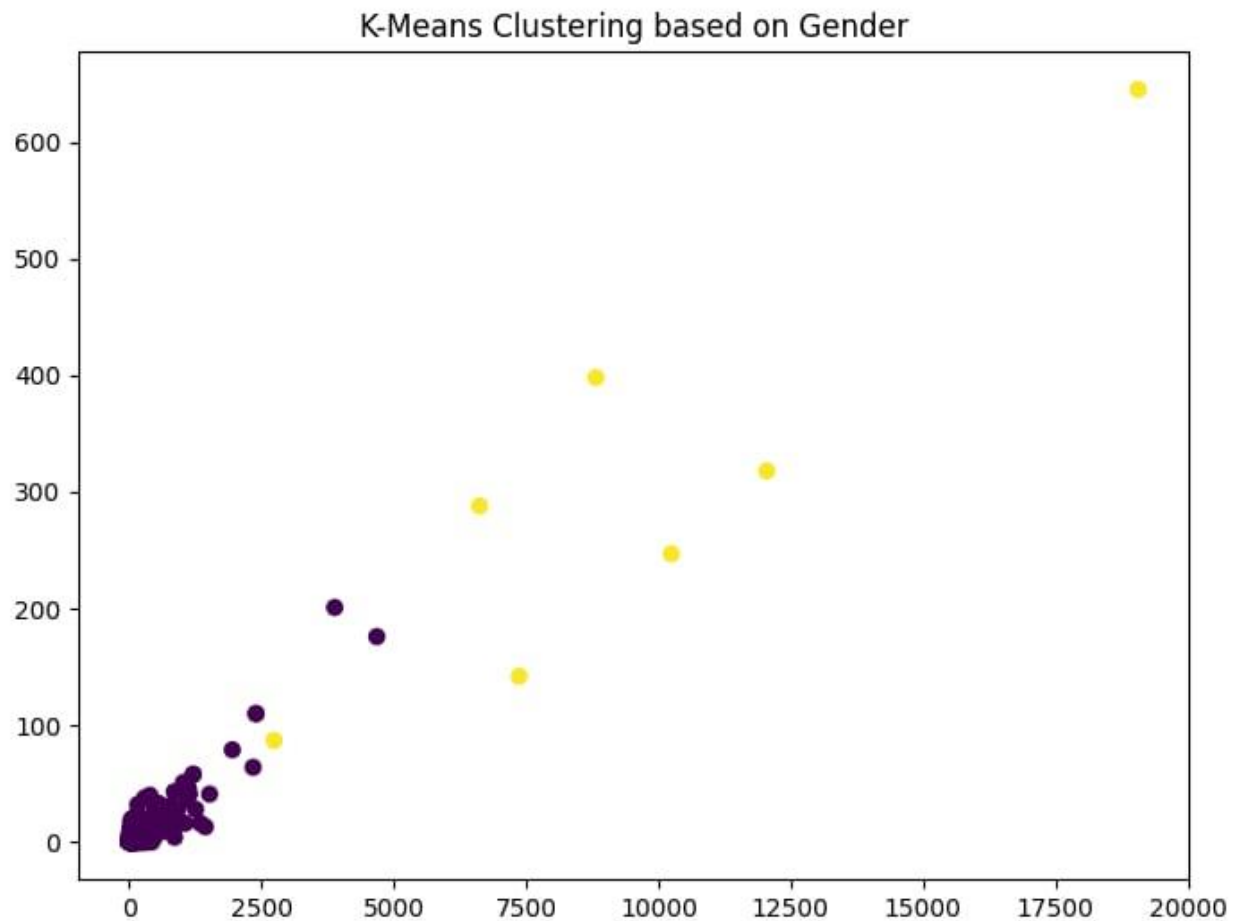

# Plot the clusters (for 2D data)

plt.figure(figsize=(8, 6))

plt.scatter(df1["Industrial Category- A- Cultivators- Persons"],
df1["Industrial Category- B- Persons"], c=labels)

plt.title('K-Means Clustering of Different Category')

plt.show()
```



```
# Plot the clusters (for 2D data)

plt.figure(figsize=(8, 6))

plt.scatter(df1["Industrial Category- H- Males"],
df1["Industrial Category- H- Females"], c=labels)

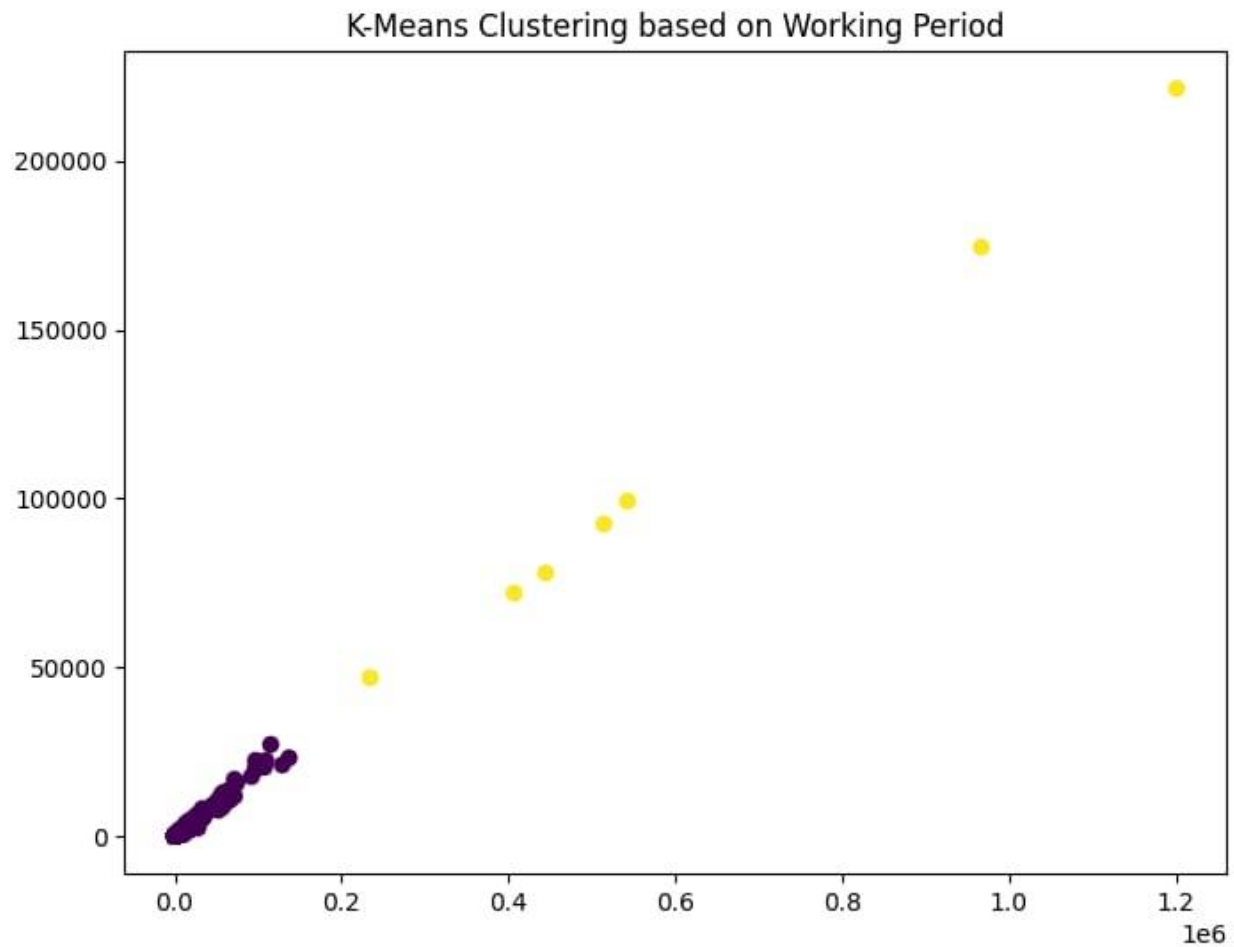
plt.title('K-Means Clustering based on Gender')

plt.show()

plt.figure(figsize=(8, 6))

plt.scatter(df1["Worked for 3 months or more but less than 6 months- Persons"],
df1["Worked for less than 3 months- Persons"], c=labels)
```

```
plt.title('K-Means Clustering based on Working Period')  
plt.show()  
  
#calculating the silhouette_score  
kmeans_labels = kmeans.fit_predict(df1)  
  
# Calculate Silhouette Score  
silhouette_avg = silhouette_score(df1, kmeans_labels)  
print(f"Silhouette Score: {silhouette_avg}")
```



```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
# Sample data
```

```
industries = ['Retail', 'Hospitality', 'Construction', 'Transportation', 'Agriculture']
```

```
margin_workers = [120, 90, 70, 80, 110]
```

```
# Creating the categorical scatter plot using Seaborn
```

```
plt.figure(figsize=(8, 6))
```

```
sns.set(style="whitegrid")
```

```
sns.scatterplot(x=industries, y=margin_workers, s=100, color='b')
```

```
# Adding title and labels
```

```
plt.title('Margin Workers in Different Industries')
```

```
plt.xlabel('Industries')
```

```
plt.ylabel('Margin Workers')
```

```
# Display the plot
```

```
plt.show()
```

Conclusion:

In conclusion, optimizing project success for Perform the demographic analysis and create visualization through best Practices for marginal worker Datasets in Tamil Nadu requires careful attention to detail and Specialized techniques. By following the best practices Outlined in this presentation, you can ensure the accuracy and Reliability of your data and improve project outcomes.