Phase 2: Innovation

Consider conducting clustering analysis to identify patterns among different industrial categories and age groups.

To conduct clustering analysis to identify patterns among different industrial categories and age groups in Python, you can use the K-Means clustering algorithm from the scikit-learn library. Here's a step-by-step guide using a given dataset

First we need to clean our data

```
import pandas as pd

# Assuming 'df' is your DataFrame
df =pd.read_csv('/content/DDW_B06_3300_State_TAMIL_NADU-2011.csv')
# Drop rows where a specific column meets a condition
df = df.drop(df[df['Age group'] == 'Total'].index)

# Drop rows based on multiple conditions
df = df.drop(df[(df['Age group'] == 'Age not stated')].index)
df = df.drop(df[df['Age group'] == '80+'].index)
df = df.drop(df[df['Age group'] == '`5-9'].index)
df = df.drop(df[df['Age group'] == '`10-14'].index)
print(df)
```

```
     Table Code State Code District Code           Area Name  \
3         B0706        `33          `000   State - TAMIL NADU
4         B0706        `33          `000   State - TAMIL NADU
5         B0706        `33          `000   State - TAMIL NADU
6         B0706        `33          `000   State - TAMIL NADU
7         B0706        `33          `000   State - TAMIL NADU
...         ...        ...           ...                 ...
1379      B0706        `33          `633  District - Tiruppur
1380      B0706        `33          `633  District - Tiruppur
1381      B0706        `33          `633  District - Tiruppur
1382      B0706        `33          `633  District - Tiruppur
1383      B0706        `33          `633  District - Tiruppur

     Total/ Rural/ Urban Age group  \
3                  Total     15-19
4                  Total     20-24
5                  Total     25-29
6                  Total     30-34
7                  Total     35-39
...                  ...       ...
1379               Urban     35-39
1380               Urban     40-49
1381               Urban     50-59
1382               Urban     60-69
1383               Urban     70-79

      Worked for 3 months or more but less than 6 months -  Persons  \
3                                                257605
4                                                478082
5                                                554851
6                                                483456
7                                                502791
...                                                 ...
1379                                               5043
1380                                               8225
1381                                               4965
1382                                               2827
1383                                                920

      Worked for 3 months or more but less than 6 months - Males  \
3                                                141262
4                                                257149
5                                                283442
6                                                240046
7                                                230695
...                                                 ...
1379                                               2455
1380                                               4269
1381                                               2800
1382                                               1590
1383                                                581

      Worked for 3 months or more but less than 6 months - Females  \
3                                                116343
4                                                220933
5                                                271409
6                                                243410
7                                                272096
```

```
df['Age_Midpoint'] = df['Age group'].apply(lambda x: sum(map(int, x.split('-'))) / 2)
```

```
df = pd.get_dummies(df, columns=['Age group'])
```

Cluster analysis can be a powerful data-mining tool for any organisation that needs to identify discrete groups of customers, sales transactions, or other types of behaviours and things. For example, insurance providers use cluster analysis to detect fraudulent claims, and banks use it for credit scoring.

Cluster analysis, like reduced space analysis (factor analysis), is concerned with data matrices in which the variables have not been partitioned beforehand into criterion versus predictor subsets.

The objective of cluster analysis is to find similar groups of subjects, where "similarity" between each pair of subjects means some global measure over the whole set of characteristics. In this article we discuss various methods of clustering and the key role that distance plays as measures of the proximity of pairs of points.

The data is then standardized using StandardScaler, and the KMeans algorithm is applied to cluster the data. The optimal number of clusters (K) is determined using the elbow method.

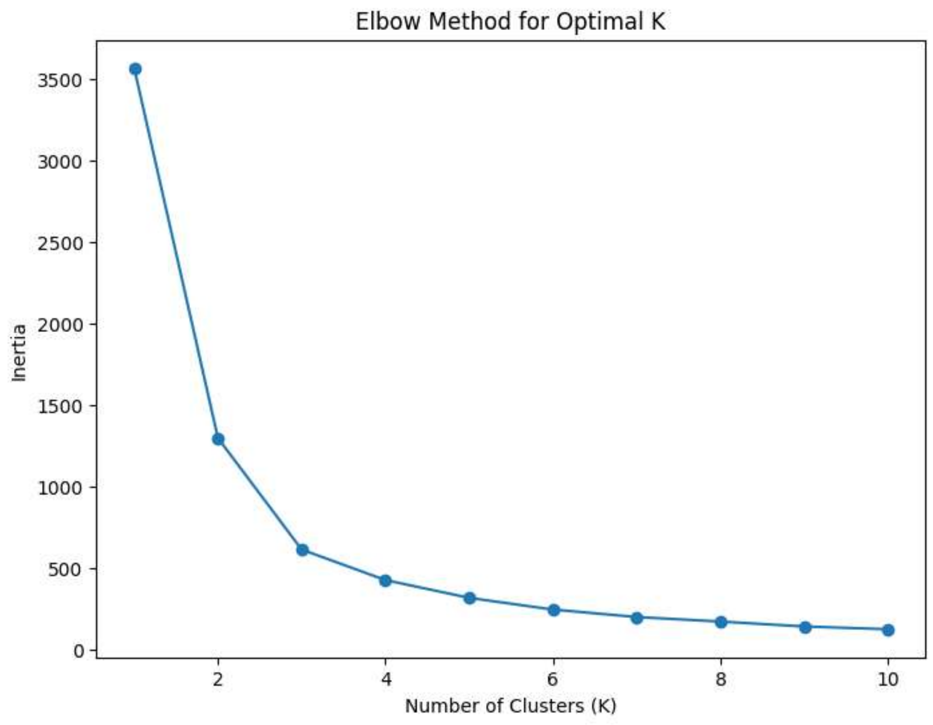
Finally, the clusters are visualized with a scatter plot where the x-axis represents age, the y-axis represents the cluster, and different industrial categories are distinguished by color.

Import necessary Libraries

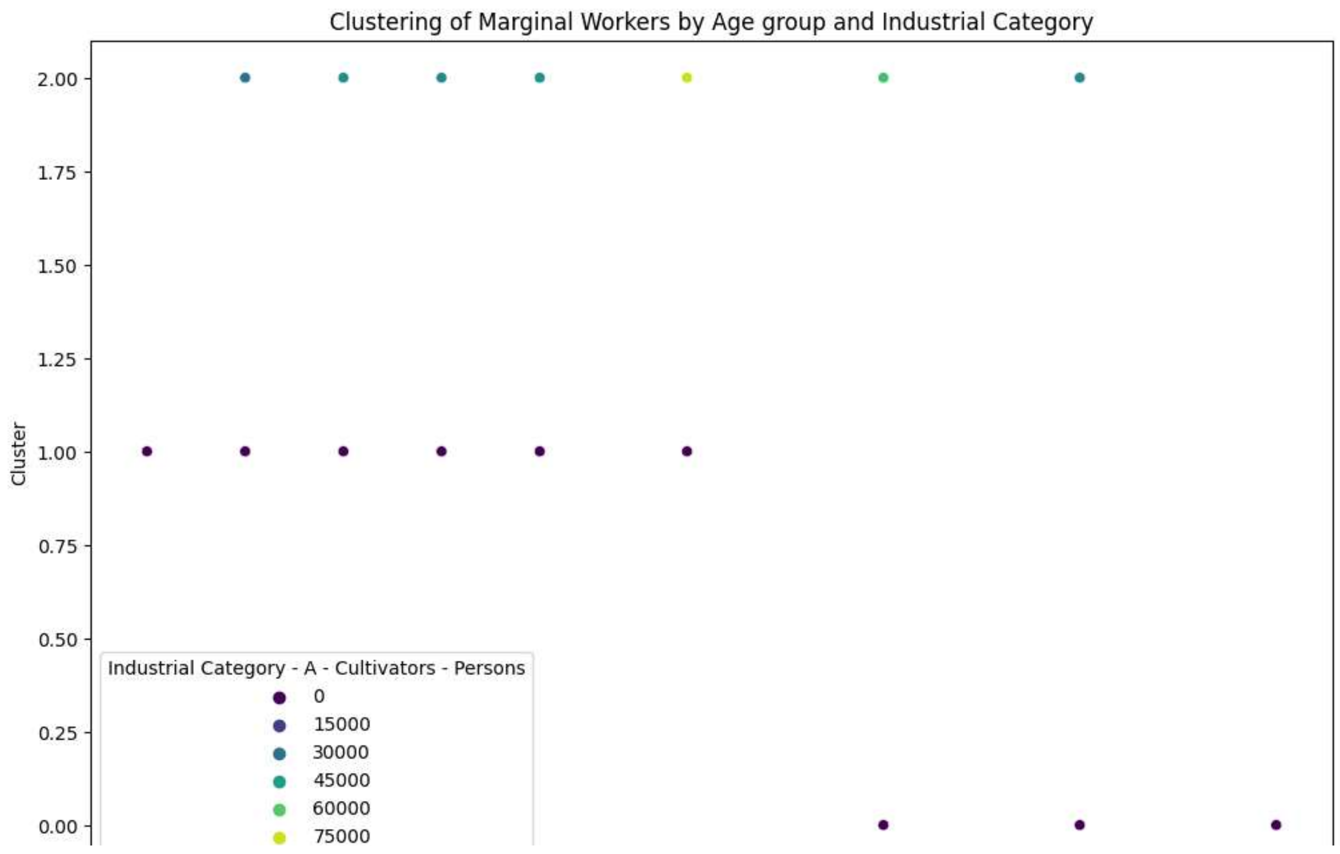
```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

```
# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[['Age_Midpoint','Industrial Category - A - Cultivators - Persons', 'Industrial Category - A - Culti

# Determine the optimal number of clusters (K) using the elbow method
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    inertia.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.show()

# Based on the elbow method, choose an appropriate value for K
k = 3

# Apply K-Means clustering
kmeans = KMeans(n_clusters=k, random_state=42)
df['cluster'] = kmeans.fit_predict(scaled_data)

# Visualize clusters
plt.figure(figsize=(12, 8))
sns.scatterplot(x='Age_Midpoint', y='cluster', hue='Industrial Category - A - Cultivators - Persons', data=df, palette='viridis')
plt.title('Clustering of Marginal Workers by Age group and Industrial Category')
plt.xlabel('Age group')
plt.ylabel('Cluster')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```



```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
```


Clustering of Marginal Workers by Age group and Industrial Category
Cluster
Industrial Category - A - Cultivators - Persons