Market Segmentation Analysis
Chapter 7: Step 5 — Extracting Segments

Tholia Priya PawanKumar

July 8, 2025

Purpose

After thorough data cleaning and exploration, Step 5 aims to divide the market into meaningful, distinct segments. The goal is to find groups of consumers with high similarity within segments and clear differences between segments.

Approaches to Extract Segments

1. Distance-Based Methods

- **Hierarchical Clustering**: Builds a dendrogram showing how data points merge. Variants include single, complete, average linkage, or Ward's method.
- **Partitioning Clustering (k-means)**: Requires pre-specifying the number of clusters. It iteratively assigns points to the nearest cluster center. Sensitive to initial conditions.
- **Self-Organizing Maps (SOMs)**: Use neural networks to project complex data onto a grid for easy interpretation.

2. Model-Based Methods

- Assume that data comes from a mixture of distributions.
- **Finite Mixture Models** and **Latent Class Analysis (LCA)** allow soft clustering with probabilities.
- Useful for mixed data types. Tools like BIC help choose the optimal number of segments.

3. Hybrid Methods

- Combine dimension reduction (e.g., PCA) with clustering.
- Example: Factor-cluster analysis. Reduces variables before clustering, useful for correlated data.

Choosing the Number of Segments

- Use elbow plots and scree plots to spot the point of diminishing returns.
- Information criteria like AIC and BIC balance fit and complexity.
- Segments must be interpretable and actionable in practice.

Validation

- Re-run clustering with different random seeds to check stability.
- Use cluster validity measures (e.g., silhouette score).
- Check if segments are distinct, interpretable, and actionable.
- Tools like gorge plots or stability plots can help.

Outputs

The result is a segment assignment for each observation, which becomes the base for profiling and targeting.

Checklist

- Choose appropriate methods and justify them.
- Select the number of segments carefully.
- Validate the solution's robustness.
- Document all steps for transparency.

Python Example

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load data
df = pd.read_csv("vacation.csv")

# Standardize numeric variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df.iloc[:, 12:])

# Apply k-means
kmeans = KMeans(n_clusters=4, random_state=42)
df['Segment'] = kmeans.fit_predict(X_scaled)

# Plot Elbow
inertia = []
for k in range(1, 11):
    km = KMeans(n_clusters=k, random_state=42)
    km.fit(X_scaled)
    inertia.append(km.inertia_)

plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```