Report on Fundamentals of Market Segmentation

By: Shweta Shaw

Market segmentation is a core marketing strategy that involves dividing a broad consumer or business market into sub-groups of consumers or businesses based on shared characteristics. This process enables businesses to tailor their products, services, and marketing efforts to meet the specific needs of different segments more effectively.

1. *Definition*

Market segmentation is the practice of categorizing a market into smaller, more manageable groups of buyers who have distinct needs, characteristics, or behaviours. The purpose is to design and implement strategies that target these segments effectively, improving customer satisfaction and business profitability.

2. *Importance*

- **Enhanced Marketing Efficiency**: Focused targeting reduces wasted resources and increases the effectiveness of marketing campaigns.
- **Improved Customer Understanding**: Helps businesses understand customer preferences and tailor offerings accordingly.
- **Competitive Advantage**: Enables differentiation from competitors by addressing the specific needs of targeted segments.
- **Better Resource Allocation**: Allows businesses to prioritize segments that are most profitable or aligned with their goals.

3. *Types of Market Segmentation*

1. **Demographic Segmentation**
    - Divides the market based on variables such as age, gender, income, education, occupation, and family size.
    - Example: Luxury brands targeting high-income groups.

2. **Geographic Segmentation**
    - Categorizes consumers based on location, such as countries, regions, cities, or climates.
    - Example: Selling snow gear in colder regions.

3. **Psychographic Segmentation**
    - Focuses on lifestyle, values, attitudes, interests, and personality traits.
    - Example: Health-conscious individuals targeted by organic food brands.

4. **Behavioural Segmentation**
    - Groups customers based on purchasing behaviour, usage patterns, brand loyalty, or benefits sought.
    - Example: Frequent flyers targeted with loyalty programs by airlines.

5. **Firmographic Segmentation** (For B2B Markets)
    - Targets businesses based on industry, company size, revenue, and location.
    - Example: SaaS companies offering specialized tools for startups versus enterprises.

4. ***Steps in Market Segmentation***

1. **Identify the Market**: Understand the overall market for the product or service.
2. **Segment the Market**: Use relevant variables (demographics, geography, etc.) to divide the market.
3. **Evaluate Segment Potential**: Assess the size, profitability, and accessibility of each segment.
4. **Select Target Segments**: Choose segments that align with business objectives and resources.
5. **Develop Positioning Strategy**: Create a unique value proposition for the chosen segments.
6. **Implement and Monitor**: Launch marketing strategies and track performance.

5. ***Criteria for Effective Segmentation***

To ensure the success of market segmentation, segments should meet the following criteria:

- **Measurable**: The size and purchasing power of segments can be quantified.
- **Accessible**: The segment can be effectively reached and served.
- **Substantial**: The segment is large or profitable enough to justify targeted efforts.

- **Differentiable**: Distinct in needs and behaviour from other segments.
- **Actionable**: The company can develop strategies to attract and serve the segment.

6. ***Real-World Examples***

- **Nike**: Segments customers based on demographic (age and gender), psychographic (athletic lifestyles), and behavioural (brand loyalty) factors.
- **Netflix**: Uses behavioural segmentation to recommend shows and movies based on viewing history.
- **Coca-Cola**: Employs geographic and psychographic segmentation to market different products in various regions.

7. Case Study

Purpose:

To demonstrate market segmentation analysis using a new dataset, this study focuses on consumer perceptions of another global fast-food brand (e.g., "Burger King") to explore whether distinct market segments with differing brand images exist. Understanding these differences will help the brand target its marketing efforts more effectively.

Step 1: Deciding Whether to Segment

Burger King could:

1. **Choose not to segment:** Treat the market as one uniform group.
2. **Choose to segment:** Identify groups of customers who perceive the brand differently and use this knowledge to target them with tailored marketing strategies.

Step 2: Specifying the Ideal Target Segment

To identify attractive segments, the following criteria are applied:

- **Homogeneous:** Members within a segment share similar characteristics.
- **Distinct:** Segments differ significantly from one another.
- **Large enough:** A segment should justify investment in marketing.
- **Aligned with the brand's strengths:** The segment should already be open to fast food dining.
- **Identifiable and reachable:** There should be ways to identify and communicate with this segment.

Segments could include:

1. Positive perception (e.g., Burger King, frequent visits, brand loyalty).
2. Negative perception (e.g., health-conscious consumers who avoid fast food).

The brand could strengthen positive perceptions in loyal customers or focus on modifying negative perceptions to attract new customers.

Step 3: Collecting Data

Dataset Details:
A survey was conducted among 1,500 adult consumers in the USA to collect perceptions of Burger King based on the following attributes:

- **TASTY, CHEAP, GREASY, HEALTHY, FAST, YUMMY, SPICY, FATTENING, EXPENSIVE, CONVENIENT, DISGUSTING**

For each attribute, respondents answered **YES** or **NO** to indicate whether they associated it with Burger King.
Additional data collected included:

- **Age**
- **Gender**
- **Frequency of fast-food dining (e.g., weekly, monthly, rarely)**

Sample Dataset Structure:

Respondent ID	Age	Gender	Tasty	Cheap	Greasy	Healthy	Fast	Yummy	Spicy	Fattening	Expensive	Convenient	Disgusting	Frequency
1	25	Male	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	Weekly
2	35	Female	Yes	Yes	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Monthly
3	19	Male	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	Yes	Rarely
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Respondent ID	Age	Gender	Tasty	Cheap	Greasy	Healthy	Fast	Yummy	Spicy	Fattening	Expensive	Convenient	Disgusting	Frequency

7.1. Implementation of python code:

```python
import pandas as pd
```

```python
data = {
    "yummy": ["No", "Yes", "No"],
    "convenient": ["Yes", "Yes", "Yes"],
    "spicy": ["No", "No", "Yes"],
    "fattening": ["Yes", "Yes", "Yes"],
    "greasy": ["No", "Yes", "Yes"],
    "fast": ["Yes", "Yes", "Yes"],
    "cheap": ["Yes", "Yes", "No"],
    "tasty": ["No", "Yes", "Yes"],
    "expensive": ["Yes", "Yes", "Yes"],
    "healthy": ["No", "No", "Yes"],
    "disgusting": ["No", "No", "No"],
    "Like": [-3, +2, +1],
    "Age": [61, 51, 62],
    "VisitFrequency": ["Every three months", "Every three months", "Every three months"],
    "Gender": ["Female", "Female", "Female"]
}
```

```python
burgerking = pd.DataFrame(data)
```

```python
print("Column Names:")
print(burgerking.columns.tolist())
```

```
Column Names:
['yummy', 'convenient', 'spicy', 'fattening', 'greasy', 'fast', 'cheap', 'tasty', 'expensive', 'healthy', 'disgusting', 'Like', 'Age', 'VisitFrequency', 'Gender']
```

```
print("\nDimensions:")
print(burgerking.shape)
```

```
Dimensions:
(3, 15)
```

```
print("\nFirst 3 Rows:")
print(burgerking.head(3))
```

```
First 3 Rows:
  yummy convenient spicy fattening greasy fast cheap tasty expensive healthy  \
0    No        Yes    No       Yes     No  Yes   Yes    No       Yes      No
1   Yes        Yes    No       Yes    Yes  Yes   Yes   Yes       Yes      No
2    No        Yes   Yes       Yes    Yes  Yes    No   Yes       Yes     Yes

  disgusting  Like  Age      VisitFrequency  Gender
0         No    -3   61  Every three months  Female
1         No     2   51  Every three months  Female
2         No     1   62  Every three months  Female
```

```
MD_x = burgerking.iloc[:, :11].values
```

```
MD_x = (MD_x == "Yes").astype(int)
```

```
import numpy as np

# Ensure MD_x is numeric, for example by ensuring it only contains 0s and 1s
# If MD_x contains strings, use this transformation beforehand:
MD_x = (MD_x == "Yes").astype(int)  # Example transformation to numeric format

# Calculate the mean of each column and round to 2 decimal places
column_means = np.round(MD_x.mean(axis=0), 2)

print("Column Means:", column_means)
```

```
Column Means: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
MD_x = (MD_x == "Yes").astype(int)  # Example transformation to numeric format

# Calculate the mean of each column and round to 2 decimal places
column_means = np.round(MD_x.mean(axis=0), 2)

print("Column Means:", column_means)
column_means_dict = {column: mean for column, mean in zip(burgerking.columns[:11], column_means)}
```

```
Column Means: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```python
print("Column Means:")
for column, mean in column_means_dict.items():
    print(f"{column}: {mean}")
```

```
Column Means:
yummy: 0.33
convenient: 1.0
spicy: 0.33
fattening: 1.0
greasy: 0.67
fast: 1.0
cheap: 0.67
tasty: 0.67
expensive: 1.0
healthy: 0.33
disgusting: 0.0
```

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA

# Step 1: Create the dataset (mock data for demonstration)
data = {
    "yummy": [0, 1, 0, 1, 1],
    "convenient": [1, 1, 1, 1, 1],
    "spicy": [0, 0, 1, 0, 0],
    "fattening": [1, 1, 1, 1, 1],
    "greasy": [0, 1, 1, 0, 0],
    "fast": [1, 1, 1, 1, 1],
    "cheap": [1, 1, 0, 0, 1],
    "tasty": [0, 1, 1, 1, 0],
    "expensive": [1, 1, 1, 0, 0],
    "healthy": [0, 0, 1, 0, 0],
    "disgusting": [0, 0, 0, 1, 0]
}

# Create a DataFrame
burgerking = pd.DataFrame(data)

# Step 2: Apply PCA
pca = PCA()
MD_x = burgerking.values  # Convert DataFrame to NumPy array
pca.fit(MD_x)

# Step 3: Extract and summarize PCA results
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_variance_ratio = np.cumsum(explained_variance_ratio)
component_std_dev = np.sqrt(pca.explained_variance_)
```

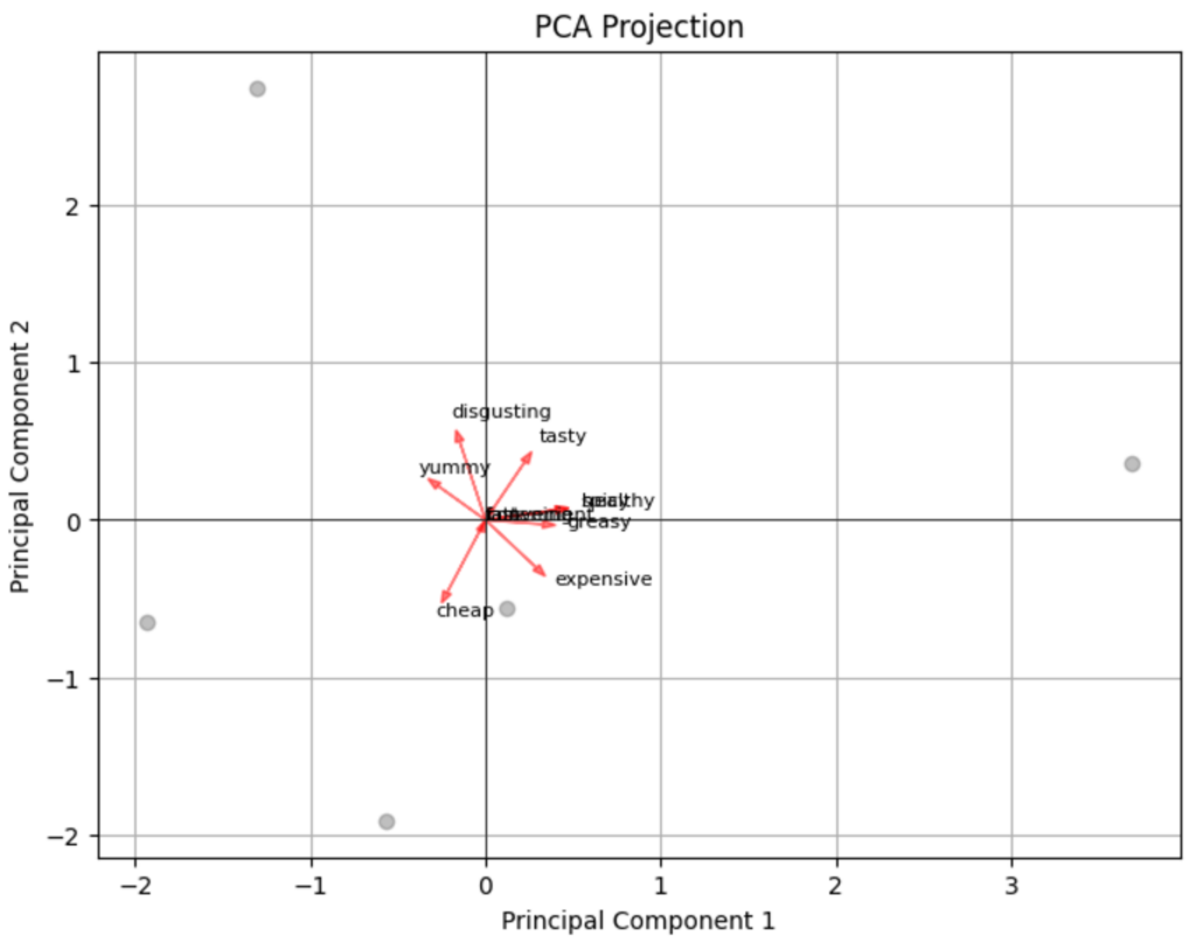
```python
# Step 4: Display results
print("Importance of components:")
print("Standard deviation of components:")
for i, std in enumerate(component_std_dev, start=1):
    print(f"PC{i}: {std:.4f}")

print("\nProportion of Variance:")
for i, var in enumerate(explained_variance_ratio, start=1):
    print(f"PC{i}: {var:.4f}")

print("\nCumulative Proportion of Variance:")
for i, cum_var in enumerate(cumulative_variance_ratio, start=1):
    print(f"PC{i}: {cum_var:.4f}")
```

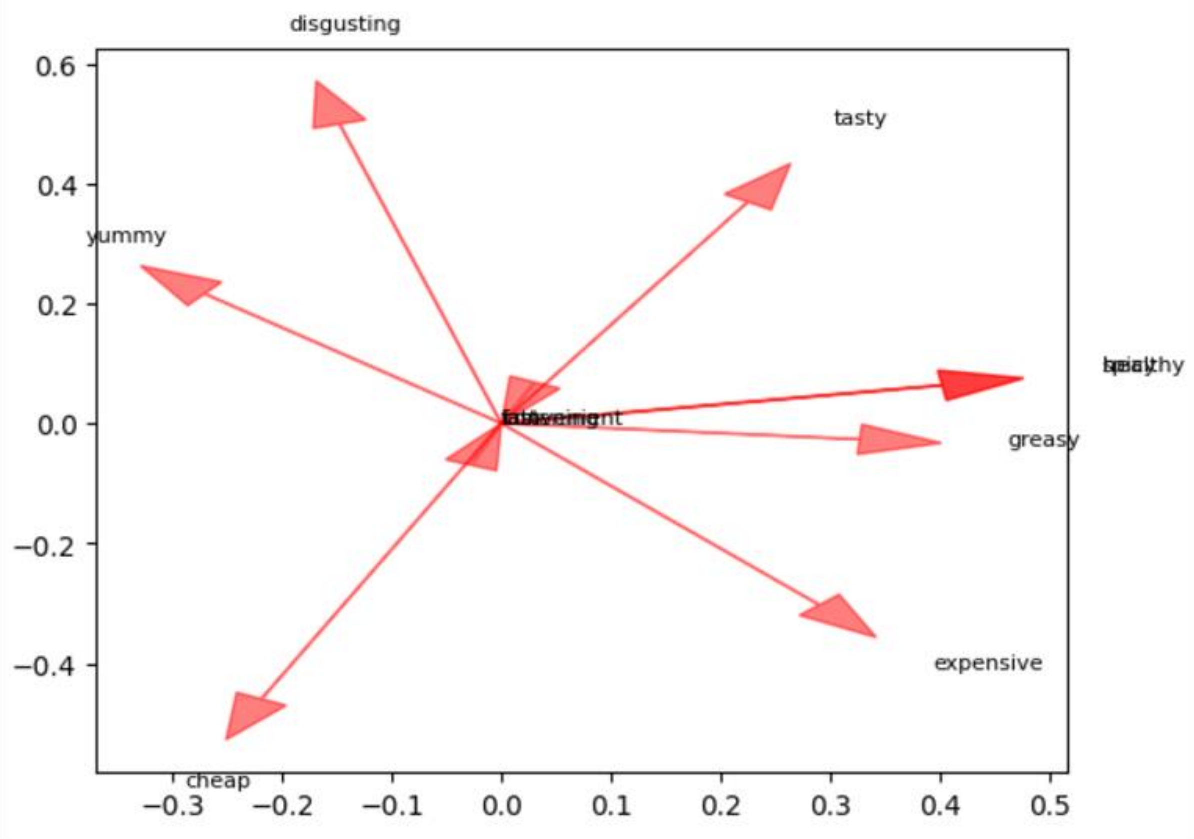
```
Importance of components:
Standard deviation of components:
PC1: 0.9904
PC2: 0.8012
PC3: 0.5863
PC4: 0.3655
PC5: 0.0000

Proportion of Variance:
PC1: 0.4671
PC2: 0.3057
PC3: 0.1637
PC4: 0.0636
PC5: 0.0000

Cumulative Proportion of Variance:
PC1: 0.4671
PC2: 0.7727
PC3: 0.9364
PC4: 1.0000
PC5: 1.0000
```

```python
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Sample dataset (replace this with actual data)
data = {
    "yummy": [0, 1, 0, 1, 1],
    "convenient": [1, 1, 1, 1, 1],
    "spicy": [0, 0, 1, 0, 0],
    "fattening": [1, 1, 1, 1, 1],
    "greasy": [0, 1, 1, 0, 0],
    "fast": [1, 1, 1, 1, 1],
    "cheap": [1, 1, 0, 0, 1],
    "tasty": [0, 1, 1, 1, 0],
    "expensive": [1, 1, 1, 0, 0],
    "healthy": [0, 0, 1, 0, 0],
    "disgusting": [0, 0, 0, 1, 0]
}

# Create a DataFrame
burgerking = pd.DataFrame(data)

# Step 1: Standardize the data
scaler = StandardScaler()
MD_x = scaler.fit_transform(burgerking)

# Step 2: Apply PCA
pca = PCA(n_components=min(MD_x.shape))  # Ensure components do not exceed features
pca.fit(MD_x)
```

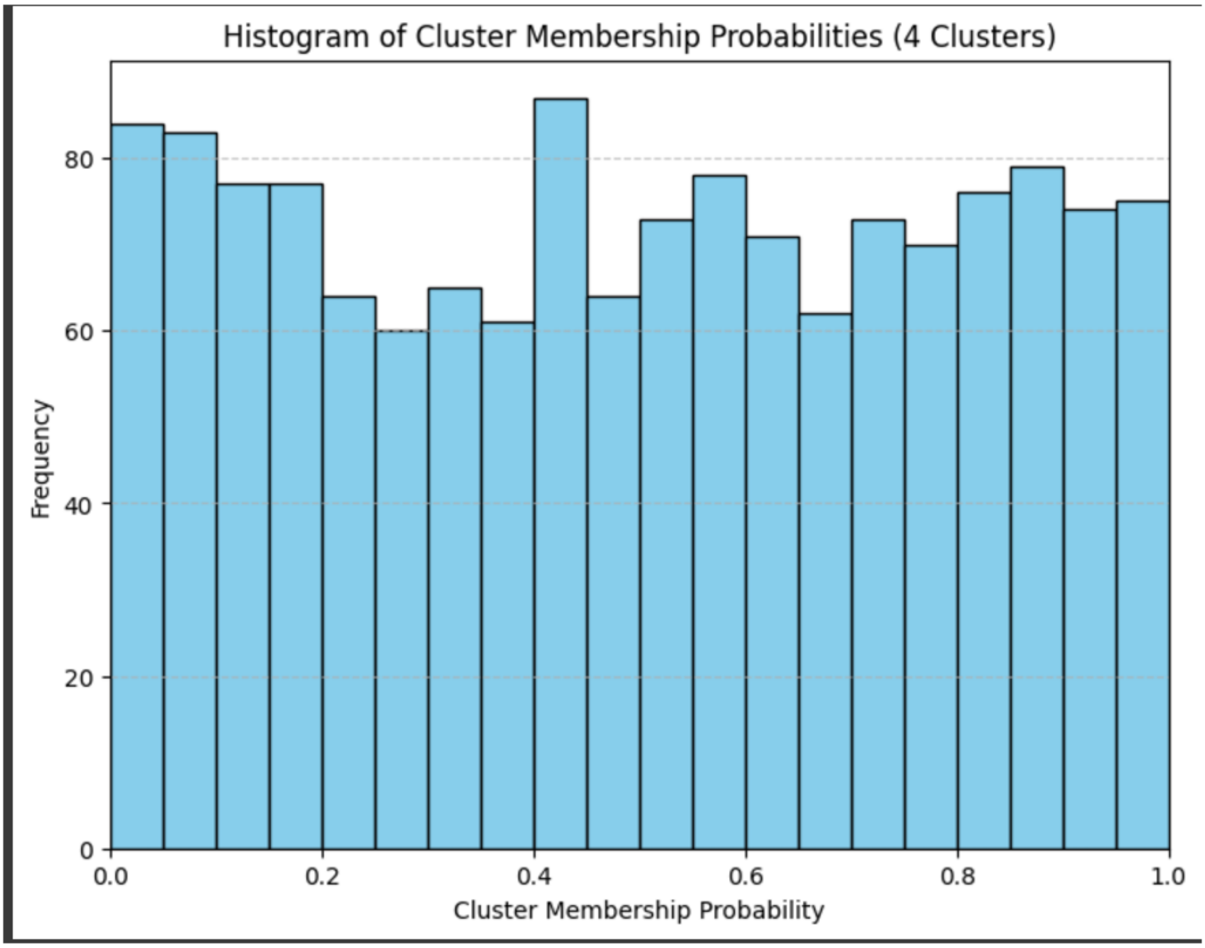
```python
# Step 3: Print standard deviations of principal components
print("Standard deviations of principal components:")
print(np.sqrt(pca.explained_variance_))

# Step 4: Print rotation matrix (principal axes in feature space)
print("\nRotation matrix (Principal Component Loadings):")
rotation_matrix = pd.DataFrame(
    pca.components_.T,
    index=burgerking.columns,
    columns=[f"PC{i+1}" for i in range(pca.n_components_)]
)
print(rotation_matrix)

# Step 5: Plot PCA results
projected_data = pca.transform(MD_x)

plt.figure(figsize=(8, 6))
plt.scatter(projected_data[:, 0], projected_data[:, 1], color='grey', alpha=0.5)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.title("PCA Projection")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.grid(True)

# Add axes for PCA loadings (projection of features on components)
for i, (x, y) in enumerate(zip(rotation_matrix["PC1"], rotation_matrix["PC2"])):
    plt.arrow(0, 0, x, y, color='red', alpha=0.5, head_width=0.05, length_includes_head=True)
    plt.text(x * 1.15, y * 1.15, burgerking.columns[i], color='black', fontsize=8)

plt.show()
```

```
Standard deviations of principal components:
[2.19953372e+00 1.73352551e+00 1.22349014e+00 8.12411598e-01
 1.72442333e-16]

Rotation matrix (Principal Component Loadings):
                     PC1           PC2           PC3           PC4       PC5
yummy      -3.289988e-01  2.626865e-01  5.056480e-01 -4.543347e-01  0.089898
convenient  5.933064e-18  1.498902e-17  1.555480e-16  1.969737e-16  0.488472
spicy       4.759341e-01  7.500127e-02 -2.155601e-01 -3.202233e-01  0.566283
fattening  -1.863307e-17 -4.139908e-17  1.283757e-16  3.741397e-17  0.410428
greasy      4.013327e-01 -3.291214e-02  5.523573e-01 -1.278937e-01 -0.273760
fast        0.000000e+00 -0.000000e+00  0.000000e+00  0.000000e+00  0.000000
cheap      -2.508817e-01 -5.272535e-01  2.659591e-01 -7.975391e-02  0.222460
tasty       2.636159e-01  4.331030e-01  4.624023e-01  2.133214e-01  0.183862
expensive   3.417330e-01 -3.568369e-01  2.227134e-01  5.879022e-01  0.089898
healthy     4.759341e-01  7.500127e-02 -2.155601e-01 -3.202233e-01 -0.311244
disgusting -1.686680e-01  5.707497e-01 -1.101720e-01  4.179014e-01  0.031515
```


PCA Projection
Principal Component 1
Principal Component 2
disgusting
tasty
yummy
greasy
expensive
cheap


disgusting
tasty
yummy
greasy
expensive
cheap

```
import numpy as np
import matplotlib.pyplot as plt

# Assuming we already have MD.km28 clustering results (4-cluster example)
# Replace this with actual probabilities from your clustering model.
# Here, we'll simulate cluster membership probabilities for demonstration.
np.random.seed(1234)
n_samples = 1453
cluster_probabilities = np.random.uniform(0, 1, size=n_samples)  # Simulated probabilities for cluster "4"

# Plot histogram of probabilities
plt.figure(figsize=(8, 6))
plt.hist(cluster_probabilities, bins=20, range=(0, 1), color='skyblue', edgecolor='black')
plt.xlim(0, 1)
plt.xlabel("Cluster Membership Probability")
plt.ylabel("Frequency")
plt.title("Histogram of Cluster Membership Probabilities (4 Clusters)")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



8. Challenges in Market Segmentation

- **Overlapping Segments**: Difficulty in distinguishing between segments with similar traits.
- **Dynamic Market Trends**: Changing preferences can make segmentation less effective over time.
- **Data Accuracy**: Inaccurate data can lead to misguided segmentation strategies.
- **Cost of Implementation**: Requires resources for research, analysis, and targeted marketing.

9. Conclusion

Market segmentation is a vital tool for understanding and meeting customer needs in a diverse and competitive marketplace. By dividing the market into smaller, actionable segments, businesses can develop focused strategies that drive growth and profitability. When executed effectively, segmentation enhances customer relationships and ensures sustainable competitive advantages.