# Market Segmentation Analysis

## Step 1: Deciding (not) to Segment

**Implications of Committing to Market Segmentation**

Market segmentation is a marketing strategy used by many companies, but it's not always the best choice. Before diving into it, organizations need to understand its implications. The main one is that it requires a long-term commitment. It's like a marriage, not a fling. Committing to segmentation means being ready to make big changes and investments. Research, surveys, product development, and advertising tailored to different segments all cost money. So, it's crucial to ensure that the expected increase in sales justifies these expenses. Market segmentation often requires significant changes like creating new products, adjusting prices, or changing how products are sold and marketed. This could also mean restructuring the organization around market segments rather than products. Given the substantial commitment involved, deciding on market segmentation should be a top-level executive decision, communicated and supported throughout the organization.

**Implementation Barriers:**

Identifying and addressing these barriers from the outset of a segmentation study is crucial. If barriers cannot be overcome, organizations may need to reconsider pursuing segmentation as a viable strategy. However, with dedication, patience, and a clear sense of purpose, organizations can navigate through these challenges to successfully implement segmentation strategies.

## Step 2: Specifying the Ideal Target Segment

**Segment Evaluation Criteria**

In market segmentation analysis, the third layer relies heavily on user input. This means that to produce useful results for an organization, user involvement must extend beyond just a briefing at the beginning or developing a marketing mix at the end. Users should be involved in most stages of the analysis, providing input throughout the process.

Once an organization commits to exploring segmentation in Step 1, it needs to contribute significantly to the analysis in Step 2. This contribution guides many subsequent steps, particularly Step 3 (data collection) and Step 8 (selecting target segments). In Step 2, the organization must establish two sets of segment evaluation criteria.

**Knock-out Criteria:** These are essential features of segments that the organization must consider targeting. These criteria are non-negotiable.

**Attractiveness Criteria:** These criteria are used to assess the relative attractiveness of remaining market segments that meet the knock-out criteria.

The literature offers various segment evaluation criteria, but it generally doesn't distinguish clearly between knock-out and attractiveness criteria. Instead, it provides a range of possible criteria at different levels of detail. These criteria help organizations determine which segments to target effectively.

**Knock-Out Criteria:**

Knock-out criteria are essential for determining if market segments resulting from segmentation analysis are suitable for further assessment using attractiveness criteria. These criteria were initially proposed by Kotler and include substantiality, measurability, and accessibility. Additional criteria have been suggested by various authors:

Homogeneity: Segment members must be similar to each other.

Distinctiveness: Segment members must be notably different from members of other segments.

Size: The segment must be large enough to justify customization of the marketing mix.

Compatibility: The segment's needs must align with the organization's strengths.

Identifiability: Segment members should be identifiable in the marketplace.

Reachability: There must be a way to connect with segment members to make customized marketing accessible to them.

These criteria are crucial and must be clearly understood by senior management, the segmentation team, and any advisory committees involved. While most criteria don't need further specification, some, like segment size, require defining the minimum viable target size. These criteria help organizations determine which segments are worth targeting and investing resources into.

**Attractiveness Criteria:**

Attractiveness criteria are additional factors that the segmentation team considers alongside knock-out criteria to evaluate the appeal of market segments. Unlike knock-out criteria, attractiveness criteria aren't binary; segments aren't simply deemed compliant or non-compliant. Instead, each segment is rated based on how well it meets specific attractiveness criteria. Segments can be more or less attractive in relation to each criterion. The collective attractiveness across all criteria influences whether a market segment is chosen as a target in Step 8 of the segmentation analysis. This means that the team assesses segments based on various factors to determine which ones are most promising for targeting.

## Step 3: Collecting Data

### Segmentation Variables

Segmentation variables are the key characteristics used to divide a sample population into market segments. In "commonsense" segmentation, typically one single characteristic, like gender, is used to create segments. For example, a sample can be split into segments of women and men based on gender. Other characteristics like age, vacation habits, and benefits sought during vacations serve as descriptor variables, helping describe the segments in detail.

In contrast, "data-driven" segmentation uses multiple variables to identify or create segments that may not be immediately obvious. For instance, instead of gender, data-driven segmentation might focus on common vacation preferences. In this case, tourists might be segmented based on shared interests like relaxation, culture, or adventure-seeking.

Both approaches rely on quality empirical data to accurately assign individuals to segments and describe those segments effectively. The data can come from various sources like surveys, observations (such as scanner data), or experimental studies. While surveys are common, they may not always reflect actual behavior accurately, especially for socially desirable actions. Therefore, it's essential to explore different data sources to ensure the data used in segmentation studies closely mirrors consumer behavior.

### Segmentation Criteria

Before segments are identified or data is collected, organizations must decide on the segmentation criterion, which is broader than the segmentation variable. While a segmentation variable refers to a specific measured value (like an item in a survey), the segmentation criterion relates to the nature of the information used for segmentation, such as geographic, demographic, psychographic, or behavioral data.

This decision cannot be easily delegated to consultants or data analysts because it requires prior market knowledge. Common segmentation criteria include geographic, demographic, psychographic, and behavioral factors.

Given the variety of segmentation criteria available, choosing the best one can be challenging. However, the recommendation is often to use the simplest approach that fits the marketing context. For instance, if demographic segmentation suffices for a product or service, it's advised to use it rather than opting for more sophisticated criteria like psychographic segmentation. The key is to use what works best for the product or service at the least possible cost.

**Geographic Segmentation**

Geographic segmentation, based on a consumer's location, is one of the oldest and simplest methods of market segmentation. For example, a national tourism organization might target tourists from neighboring countries, recognizing language differences as a pragmatic reason to treat them as separate segments. Global companies like Amazon and IKEA adapt their offerings based on customers' geographic location.

The main advantage of geographic segmentation is its simplicity. It's easy to assign consumers to geographic units, making it straightforward to target communication messages and select appropriate channels.

However, a significant drawback is that sharing the same geographic area doesn't guarantee similarity in other relevant characteristics, such as purchasing preferences. For instance, residents of luxury suburbs may have varied preferences beyond their location. In tourism, people from the same country may have diverse ideal holidays based on factors like family status or interests.

**Socio-Demographic Segmentation**

Socio-demographic segmentation involves dividing a market based on factors like age, gender, income, and education. These segments can be valuable in various industries. For instance, luxury goods often target high-income individuals, while cosmetics cater to different genders. Socio-demographic factors also play a role in sectors like baby products, retirement villages, and tourism.

Similar to geographic segmentation, socio-demographic criteria make it easy to determine segment membership for each consumer. Sometimes, socio-demographic factors can explain specific product preferences. For example, having children may influence families to choose family-oriented vacation destinations.

However, socio-demographic criteria may not always provide sufficient insight for segmentation decisions. Research suggests that demographics explain only about 5% of consumer behavior variance. Some experts argue that values, tastes, and preferences are more influential in consumers' buying decisions compared to socio-demographic factors. Therefore, while socio-demographic segmentation can be useful, it's essential to consider other factors like values and preferences for more effective segmentation.

**Psychographic Segmentation**

Psychographic segmentation involves grouping people based on psychological criteria such as beliefs, interests, preferences, aspirations, or the benefits they seek when purchasing a product. It's a more intricate approach compared to geographic or socio-demographic segmentation because it aims to capture the complexities of individuals' minds.

Psychographic segmentation studies typically use multiple variables to capture various aspects of individuals' psychographic profiles. For instance, in tourism, variables like travel motives or perceived risks during vacations may be considered.

One advantage of the psychographic approach is that it often provides deeper insights into the underlying reasons for differences in consumer behavior. For example, tourists motivated by cultural exploration are likely to prefer destinations with rich cultural offerings. However, this approach's complexity lies in accurately determining segment memberships for consumers and ensuring the reliability and validity of the measures used to capture psychographic dimensions.

**Behavioural Segmentation**

Behavioral segmentation involves grouping individuals based on their actual behaviors or reported behaviors. This approach considers various behaviors such as prior product experience, purchase frequency, amount spent on purchases, and information search behavior.

One advantage of behavioral segmentation is that it uses the behavior of interest as the basis for segment extraction. This means that individuals are grouped based on the behavior that matters most, providing valuable insights into consumer preferences and tendencies. For example, studies have shown that behavioral variables reported by tourists can be more effective than geographic variables in segmentation analysis.

Using behavioral data can also eliminate the need to develop measures for psychological constructs. However, one challenge of behavioral segmentation is that behavioral data may not always be readily available, especially when aiming to include potential customers who have not previously purchased the product. This limitation may restrict the segmentation analysis to existing customers of the organization.

## Step 6: Profiling Segments

**Identifying Key Characteristics of Market Segments**

The purpose of the profiling step is to understand the characteristics of the market segments identified during the extraction process. This step is essential in data-driven market segmentation, as the defining characteristics of resulting segments are unknown until after the analysis. Unlike in commonsense segmentation, where segment profiles are predefined (e.g., age groups if age is the segmentation variable), in data-driven segmentation, the characteristics of segments need to be identified based on the segmentation variables used.

Profiling involves characterizing each market segment individually and comparing them to each other. For example, if winter tourists in Austria predominantly engage in alpine skiing, this activity may characterize a segment, but it may not differentiate that segment from others.

During profiling, multiple alternative segmentation solutions are inspected, especially if natural segments don't exist in the data, and a reproducible or constructive segmentation approach is required. Effective profiling is crucial for interpreting segments correctly, which is essential for making strategic marketing decisions.

Data-driven segmentation solutions can be challenging to interpret, and many managers struggle with understanding them. Studies have shown that a significant percentage of marketing managers find segmentation analysis like a "black box," indicating difficulties in grasping the insights provided by the segmentation results. Therefore, clear and insightful presentation of segmentation results is crucial for aiding managers in making informed marketing decisions.

**Segment Profiling with Visualisations**

Visualizations play a crucial role in segment profiling, especially in data-driven market segmentation. While traditional tabular representations are common, they often fail to fully utilize the power of graphics, despite the integral role of data visualization in statistical analysis.

Graphics are particularly valuable in exploratory statistical analysis, such as cluster analysis, as they provide insights into complex relationships between variables. Moreover, in the era of big data, visualization offers a simple way to monitor developments over time. Experts recommend the use of visualization techniques to enhance the interpretation of market segmentation analysis results, as they can make insights more accessible and intuitive.

Research has shown that graphical representations are more insightful than tabular forms, making it easier to interpret results. Various visualization techniques are available for cluster analysis and mixture models, aiding in the interpretation of segmentation solutions.

# CODE

```python
In [143]: import numpy as np
          import pandas as pd
          from sklearn.decomposition import PCA
          from sklearn.cluster import KMeans
          from scipy.cluster.hierarchy import linkage, dendrogram
          from sklearn.mixture import GaussianMixture
          from statsmodels.formula.api import ols
          import matplotlib.pyplot as plt
          from sklearn.tree import DecisionTreeClassifier, plot_tree
```

```python
In [144]: # Load the data
          mcdonalds = pd.read_csv('C:/Users/bavit/Downloads/mcdonalds.csv')
```

```python
In [145]: # Explore the data
          print(mcdonalds.columns)
          print(mcdonalds.shape)
          print(mcdonalds.head(3))
```

```python
In [146]: # Preprocess the data
          MD_x = (mcdonalds.iloc[:, 0:11] == 'Yes').astype(int)
          print(MD_x.mean().round(2))
```

```
yummy         0.55
convenient    0.91
spicy         0.09
fattening     0.87
greasy        0.53
fast          0.90
cheap         0.60
tasty         0.64
expensive     0.36
healthy       0.20
disgusting    0.24
dtype: float64
```

```
Index(['yummy', 'convenient', 'spicy', 'fattening', 'greasy', 'fast', 'cheap',
       'tasty', 'expensive', 'healthy', 'disgusting', 'Like', 'Age',
       'VisitFrequency', 'Gender'],
      dtype='object')
(1453, 15)
  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
```

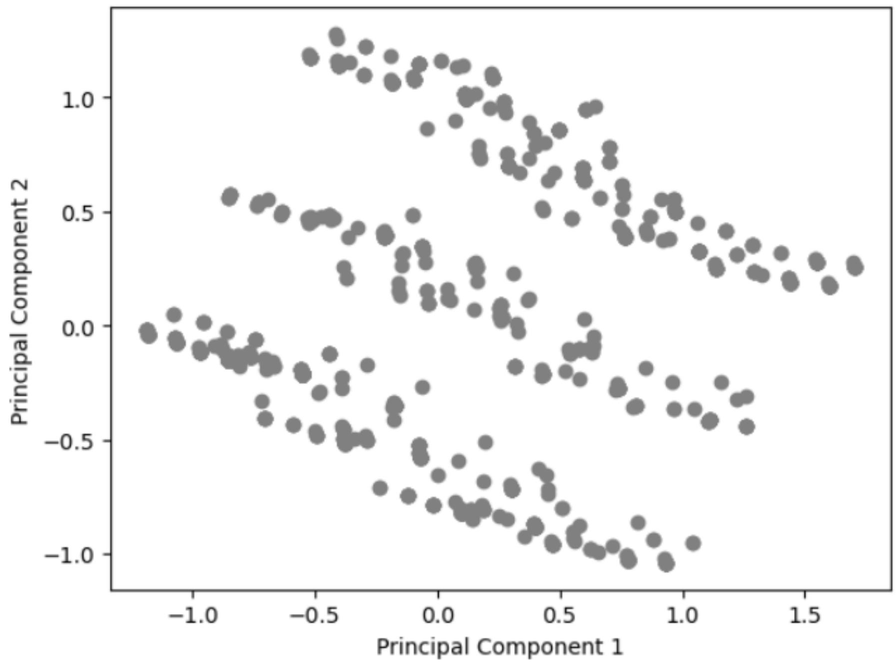
In [147]:

```python
# Principal Component Analysis
MD_pca = PCA(n_components=len(MD_x.columns))
MD_pca.fit(MD_x)
print(MD_pca.explained_variance_ratio_)
print(MD_pca.components_.round(1))
```

```
[0.29944723 0.19279721 0.13304535 0.08309578 0.05948052 0.05029956
 0.0438491  0.03954779 0.0367609  0.03235329 0.02932326]
[[-0.5 -0.2 -0.   0.1  0.3 -0.1 -0.3 -0.5  0.3 -0.2  0.4]
 [ 0.4  0.   0.  -0.  -0.1 -0.1 -0.6  0.3  0.6  0.1 -0.1]
 [-0.3 -0.1 -0.  -0.3 -0.8 -0.1 -0.1 -0.3  0.   0.2 -0.1]
 [ 0.1 -0.1  0.2 -0.4  0.3 -0.1  0.1 -0.   0.1  0.8  0.4]
 [-0.3  0.3  0.1 -0.1  0.4  0.1 -0.1 -0.2 -0.   0.3 -0.7]
 [ 0.2 -0.3 -0.4 -0.4  0.2 -0.6 -0.1 -0.1 -0.3 -0.2 -0.2]
 [-0.3 -0.1  0.7 -0.4  0.  -0.1 -0.   0.4 -0.1 -0.3 -0. ]
 [ 0.  -0.1  0.4  0.6 -0.1 -0.6  0.1 -0.1  0.   0.2 -0.2]
 [ 0.6 -0.   0.4 -0.2 -0.   0.2  0.1 -0.6  0.1 -0.2 -0.1]
 [-0.1 -0.7 -0.1 -0.   0.   0.2  0.4  0.1  0.5 -0.  -0.3]
 [ 0.  -0.5  0.1  0.3  0.   0.3 -0.5  0.  -0.5  0.2 -0. ]]
```

In [148]:

```python
# Visualize the PCA
plt.figure()
plt.scatter(MD_pca.transform(MD_x)[:, 0], MD_pca.transform(MD_x)[:, 1], color='grey')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```



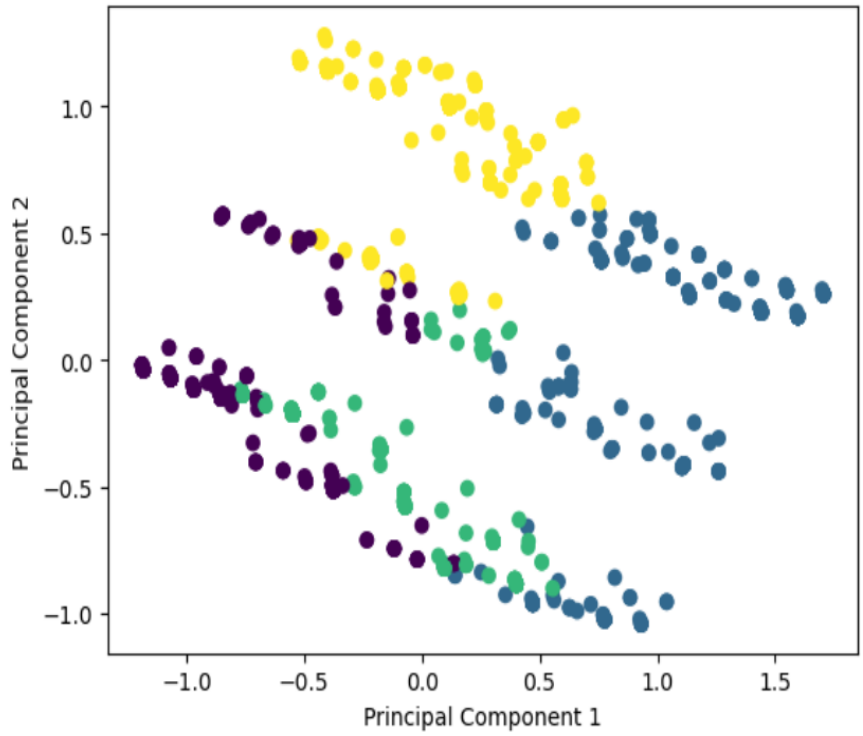

In [149]:

```python
# K-Means Clustering

#from sklearn.cluster import KMeans
#for k in range(2, 9):
#    km = KMeans(n_clusters=k, random_state=1234).fit(MD_x)
#    print(f'For k={k}, Silhouette Score: {km.score(MD_x):.3f}')

#MD_km4 = KMeans(n_clusters=4, random_state=1234).fit(MD_x)
#print(MD_km4.labels_)
# Open Command Prompt or Anaconda Prompt
# Set the OMP_NUM_THREADS environment variable
#set OMP_NUM_THREADS=6

from sklearn.cluster import MiniBatchKMeans

# Set the number of clusters to explore

for k in range(2, 9):
    km = MiniBatchKMeans(n_clusters=k, random_state=1234, batch_size=100).fit(MD_x)
    print(f'For k={k}, Silhouette Score: {km.score(MD_x):.3f}')

# Fit MiniBatchKMeans with 4 clusters
MD_km4 = MiniBatchKMeans(n_clusters=4, random_state=1234, batch_size=100).fit(MD_x)
print(MD_km4.labels_)
```

```
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=2, Silhouette Score: -2211.320
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=3, Silhouette Score: -1923.375
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=4, Silhouette Score: -1664.343
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=5, Silhouette Score: -1522.653
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=6, Silhouette Score: -1478.870
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=7, Silhouette Score: -1329.783
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
For k=8, Silhouette Score: -1248.535
C:\Users\bavit\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1848: UserWarning: MiniBatchKMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can prevent it by setting batch_size >= 2048 or by setting the environment variable OMP_NUM_THREADS=1
  warnings.warn(
[1 3 3 ... 3 0 1]
```

In [150]:

```python
# Visualize the clusters
plt.figure()
plt.scatter(MD_pca.transform(MD_x)[:, 0], MD_pca.transform(MD_x)[:, 1], c=MD_km4.labels_)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```



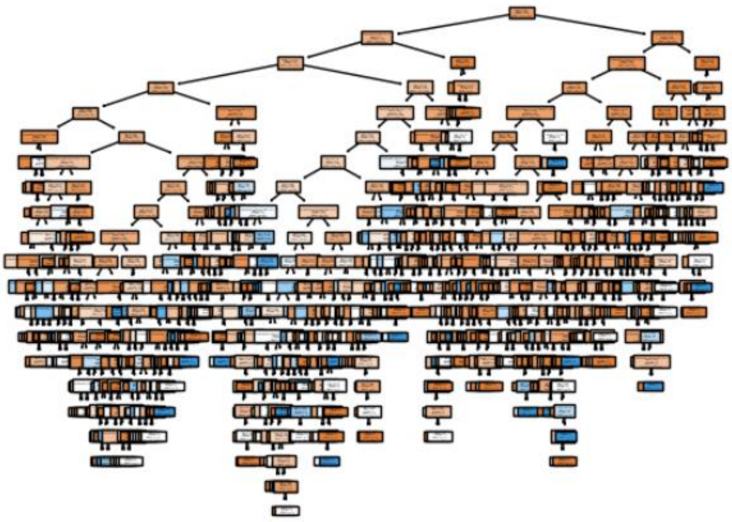
```
In [151]: # Gaussian Mixture Model
from sklearn.mixture import GaussianMixture
for k in range(2, 9):
    gm = GaussianMixture(n_components=k, random_state=1234).fit(MD_x)
    print(f'For k={k}, BIC: {gm.bic(MD_x):.3f}')

MD_gm4 = GaussianMixture(n_components=4, random_state=1234).fit(MD_x)
print(MD_gm4.means_)
print(MD_gm4.covariances_)
```

```
For k=8, BIC: -43486.202
[[0.069869   0.6768559  0.08296943 0.89956332 0.71179039 0.70742358
  0.10043668 0.11353712 0.97816594 0.06550218 0.65502183]
 [1.         0.91133006 0.09852216 0.3842365  0.25615766 0.7536946
  0.62068967 0.79310346 0.31527093 0.48768472 0.        ]
 [0.         0.85906041 0.06711409 0.84899329 0.5704698  0.90604027
  0.81543624 0.08724834 0.         0.11073825 0.40939598]
 [0.80774551 1.         0.10650069 1.         0.52558782 1.
  0.66113416 1.         0.32088521 0.19640387 0.11203319]]
[[[ 6.49883192e-02  9.47731737e-03  2.93663355e-03 -6.08302666e-03
    1.14032913e-02 -1.01256651e-02  1.91834633e-02  3.57353980e-02
   -2.03085372e-02  4.15705268e-03  2.41032779e-02]
  [ 9.47731737e-03  2.18722992e-01  1.37106463e-02 -6.25464808e-03
   -3.19978643e-02  3.20932095e-02 -6.84578860e-03  1.04879770e-02
   -2.68873591e-03  1.24330200e-02 -7.65431628e-02]
  [ 2.93663355e-03  1.37106463e-02  7.60865056e-02 -1.35008867e-02
   -2.28828588e-03  6.80765050e-03  4.00450029e-04  3.68032646e-03
   -6.92206480e-03  7.66575771e-03 -1.94504300e-03]
  [-6.08302666e-03 -6.25464808e-03 -1.35008867e-02  9.03501543e-02
```

```
In [152]: # Regression

import pandas as pd
from statsmodels.formula.api import ols

# Remove non-numeric characters from 'Like' column and convert to int
mcdonalds['Like'] = mcdonalds['Like'].str.replace(r'\D', '', regex=True).astype(int)

# Create the dependent variable 'Like_n'
mcdonalds['Like_n'] = 6 - mcdonalds['Like']

# Construct the formula for regression
formula = 'Like_n ~ ' + ' + '.join(mcdonalds.columns[:11])

# Fit the regression model
model = ols(formula, data=mcdonalds).fit()

# Print the summary of the regression results
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Like_n   R-squared:                       0.146
Model:                            OLS   Adj. R-squared:                  0.140
Method:                 Least Squares   F-statistic:                     22.44
Date:                Fri, 05 Apr 2024   Prob (F-statistic):           8.50e-43
Time:                        19:51:24   Log-Likelihood:                -2670.2
No. Observations:                1453   AIC:                             5364.
Df Residuals:                    1441   BIC:                             5428.
Df Model:                          11
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
Intercept               2.9172      0.222     13.127      0.000       2.481       3.353
yummy[T.Yes]           -0.9784      0.114     -8.580      0.000      -1.202      -0.755
convenient[T.Yes]       1.1243      0.154      7.315      0.000       0.823       1.426
spicy[T.Yes]            0.1988      0.140      1.423      0.155      -0.075       0.473
fattening[T.Yes]        0.3091      0.132      2.344      0.019       0.050       0.568
greasy[T.Yes]           0.1061      0.089      1.187      0.235      -0.069       0.281
fast[T.Yes]            -0.1117      0.142     -0.787      0.432      -0.390       0.167
cheap[T.Yes]           -0.3134      0.122     -2.575      0.010      -0.552      -0.075
tasty[T.Yes]            0.2682      0.120      2.238      0.025       0.033       0.503
expensive[T.Yes]       -0.2374      0.124     -1.908      0.057      -0.482       0.007
healthy[T.Yes]         -0.3392      0.112     -3.035      0.002      -0.558      -0.120
disgusting[T.Yes]      -0.9448      0.114     -8.258      0.000      -1.169      -0.720
==============================================================================
Omnibus:                       83.831   Durbin-Watson:                   1.913
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               41.479
Skew:                           0.228   Prob(JB):                     9.84e-10
Kurtosis:                       2.309   Cond. No.                         14.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [153]:
```
# Decision Tree

from sklearn.preprocessing import OneHotEncoder

# Convert categorical variables to one-hot encoding
X_encoded = pd.get_dummies(X, columns=['VisitFrequency', 'Gender'])

# Fit the decision tree classifier
tree.fit(X_encoded, y)

# Plot the decision tree
plot_tree(tree, feature_names=X_encoded.columns, class_names=['Not Segment 3', 'Segment 3'], filled=True)
plt.show()
```



In [154]:
```
# Visualize the segments
plt.figure()
plt.scatter(MD_pca.transform(MD_x)[:, 0], MD_pca.transform(MD_x)[:, 1], c=MD_km4.labels_)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```