Market Segmentation Analysis

Case Study: Fast Food

**Step 4: Exploring Data**

import pandas as pd

from pydataset import data

mcdonalds = data('mcdonalds')

print(list(mcdonalds.columns))

print(mcdonalds.shape)

print(mcdonalds.head(3))

2) import pandas as pd

from sklearn.decomposition import PCA

# assuming that MD.x is a pandas DataFrame

MD\_pca = PCA().fit(MD.x)

print('Standard deviations (1, .., p=11):')

print(MD\_pca.explained\_variance\_)

print('Proportion of Variance:')

print(MD\_pca.explained\_variance\_ratio\_)

print('Cumulative Proportion:')

print(np.cumsum(MD\_pca.explained\_variance\_ratio\_))

print('Factor Loadings:')

print(pd.DataFrame(MD\_pca.components\_).T.round(1))

4) import pandas as pd

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

import numpy as np

# Load data

MD = pd.read\_csv("MD.csv")

MD.x = MD.drop(['Segment'], axis=1)

# PCA

MD\_pca = PCA().fit(MD.x)

print("Importance of components:")

print(pd.DataFrame({'Standard deviation': np.sqrt(MD\_pca.explained\_variance\_),

'Proportion of Variance': MD\_pca.explained\_variance\_ratio\_,

'Cumulative Proportion': np.cumsum(MD\_pca.explained\_variance\_ratio\_)}))

# Factor Loadings

print("Factor Loadings:")

print(pd.DataFrame(MD\_pca.components\_.T, columns=['PC{}'.format(i+1) for i in range(MD\_pca.n\_components\_)], index=MD.x.columns))

# Plot projected data and factor loadings

from sklearn.preprocessing import StandardScaler

# Project data into PC space

MD\_pca\_proj = MD\_pca.transform(StandardScaler().fit\_transform(MD.x))

# Plot projected data

plt.scatter(MD\_pca\_proj[:, 0], MD\_pca\_proj[:, 1], color='grey')

# Add factor loadings as arrows

scale = 1.0

for i, v in enumerate(MD\_pca.components\_.T):

plt.arrow(0, 0, scale\*v[0], scale\*v[1], color='r', width=0.005, head\_width=0.05, length\_includes\_head=True)

plt.text(scale\*v[0]\*1.1, scale\*v[1]\*1.1, MD.x.columns[i], color='r')

plt.xlabel('PC1')

plt.ylabel('PC2')

plt.title('Projected data and factor loadings')

plt.show()

**Using k-Means**

import numpy as np

from sklearn.cluster import KMeans

# Load the data

data = np.loadtxt("my\_data.csv", delimiter=",")

# Normalize the data

data = (data - np.mean(data, axis=0)) / np.std(data, axis=0)

# Cluster the data using the k-means algorithm with 2 to 8 clusters

inertias = []

for k in range(2, 9):

kmeans = KMeans(n\_clusters=k, n\_init=10, random\_state=1234).fit(data)

inertias.append(kmeans.inertia\_)

# Plot the elbow curve

import matplotlib.pyplot as plt

plt.plot(range(2, 9), inertias, marker="o")

plt.xlabel("Number of clusters")

plt.ylabel("Inertia")

plt.show()

# Choose the number of clusters based on the elbow curve and re-fit the model

kmeans = KMeans(n\_clusters=4, n\_init=10, random\_state=1234).fit(data)

# Print the cluster labels

print(kmeans.labels\_)

**Using Mixtures of Distributions**

# Load required packages

import rpy2.robjects as robjects

from rpy2.robjects.packages import importr

import rpy2.robjects.numpy2ri

rpy2.robjects.numpy2ri.activate()

# Load data (assuming it's in a file called "MD.x")

from rpy2.robjects import r

r("data <- read.table('MD.x')")

# Load "flexmix" package and FLXMCmvbinary function

flexmix = importr('flexmix')

FLXMCmvbinary = robjects.r('FLXMCmvbinary')

# Set seed and run stepFlexmix function

r('set.seed(1234)')

MD\_m28 = flexmix.stepFlexmix("MD.x ~ 1", k=robjects.IntVector(range(2,9)), nrep=10, model=FLXMCmvbinary(), verbose=False)

# Print results

print(MD\_m28)

# Plot information criteria

graphics = importr('graphics')

graphics.plot(MD\_m28, ylab="value of information criteria (AIC, BIC, ICL)")

**Using Mixtures of Regression Models**

like\_codes = {'I HATE IT!': -5, '-4': -4, '-3': -3, '-2': -2, '-1': -1, '0': 0, '+1': 1, '+2': 2, '+3': 3, '+4': 4, 'I LOVE IT!': 5}

mcdonalds['Like.n'] = mcdonalds['Like'].apply(lambda x: 6 - like\_codes[x])

from sklearn.mixture import GaussianMixture

X = mcdonalds[['yummy', 'convenient', 'spicy', 'fattening', 'greasy', 'fast', 'cheap', 'tasty', 'expensive', 'healthy', 'disgusting']]

y = mcdonalds['Like.n']

model = GaussianMixture(n\_components=2, covariance\_type='full', random\_state=1234, n\_init=10)

model.fit(X, y)

from sklearn.mixture import GaussianMixture

X = mcdonalds[['yummy', 'convenient', 'spicy', 'fattening', 'greasy', 'fast', 'cheap', 'tasty', 'expensive', 'healthy', 'disgusting']]

y = mcdonalds['Like.n']

model = GaussianMixture(n\_components=2, covariance\_type='full', random\_state=1234, n\_init=10)

model.fit(X, y)

**Describing Segments**

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.graphics.mosaicplot import mosaic

# Load the fast food dataset

df = pd.read\_csv("fastfood.csv")

# Extract the segment membership for each consumer for the four-segment solution

k4 = df['Segment']

# Cross-tabulate segment membership and the love-hate variable

table = pd.crosstab(k4, df['LoveHate'])

# Generate the mosaic plot with cells colours indicating the deviation of the observed frequencies

mosaic(table, gap=0.01, title="",

properties=lambda key: {'color': 'r' if 'I hate it!' in key else 'b'})

plt.xlabel("Segment number")

plt.ylabel("Love-Hate for McDonald's")

plt.show()

**Selecting (the) Target Segment(s)**

import pandas as pd

import matplotlib.pyplot as plt

# Load data

mcdonalds = pd.read\_csv('mcdonalds\_data.csv')

# Compute mean values for each segment

visit = mcdonalds.groupby('Segment')['VisitFrequency'].mean()

like = mcdonalds.groupby('Segment')['Like.n'].mean()

female = mcdonalds.groupby('Segment')['Gender'].apply(lambda x: (x=='Female').mean())

# Create plot

plt.scatter(visit, like, s=female\*1000, alpha=0.5)

plt.xlim(2, 4.5)

plt.ylim(-3, 3)

plt.xlabel('Frequency of Visiting McDonald\'s')

plt.ylabel('Extent of Liking McDonald\'s')

for i, seg in enumerate(visit.index):

plt.text(visit[seg], like[seg], seg, ha='center', va='center')

plt.show()