Aprendizagem 2024
Homework IV - Group 92
Maria Maló - ist102994
André Feliciano - ist100286

Part I: Pen and paper

1.

$$\mathbf{x_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x_2} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad \mathbf{x_3} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$\mathbf{u_1} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \quad \mathbf{u_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The priors are:

$$\pi_1 = 0.5, \quad \pi_2 = 0.5$$

The joint probabilities are the priors times the likelihoods:

$$\gamma_k(x_i) = \frac{\pi_k \mathcal{N}(x_i \mid \mathbf{u_k}, \Sigma_k)}{\sum_{j=1}^{2} \pi_j \mathcal{N}(x_i \mid \mathbf{u_j}, \Sigma_j)}$$

For the first Epoch:

x_1 joint probabilities:

$$\gamma_1(x_1) = \frac{\mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1)}{\mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.319$$

For a 2-dimensional Gaussian distribution (d = 2):

$$\mathcal{N}(x_i \mid \mathbf{u}, \Sigma) = \frac{1}{(2\pi)|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mathbf{u})^T \Sigma^{-1} (x_i - \mathbf{u})\right)$$

To speed up the calculations, we used the SciPy package `multivariate_normal` to compute this $\mathcal{N}(x_i \mid \mathbf{u}, \Sigma)$ for each observation and cluster:

$$\mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1) \approx 0.029, \quad \mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2) \approx 0.062$$

$$\gamma_2(x_1) = \frac{\mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2)}{\mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.681$$

x_2 joint probabilities:

$$\gamma_1(x_2) = \frac{\mathcal{N}(x_2 \mid \mathbf{u_1}, \Sigma_1)}{\mathcal{N}(x_2 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_2 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.094$$

$$\mathcal{N}(x_2 \mid \mathbf{u_1}, \Sigma_1) \approx 0.005, \quad \mathcal{N}(x_2 \mid \mathbf{u_2}, \Sigma_2) \approx 0.048$$

$$\gamma_2(x_2) = \frac{\mathcal{N}(x_2 \mid \mathbf{u_2}, \Sigma_2)}{\mathcal{N}(x_2 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_2 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.906$$

x_3 joint probabilities:

$$\gamma_1(x_3) = \frac{\mathcal{N}(x_3 \mid \mathbf{u_1}, \Sigma_1)}{\mathcal{N}(x_3 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_3 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.766$$

$$\mathcal{N}(x_3 \mid \mathbf{u_1}, \Sigma_1) \approx 0.036, \quad \mathcal{N}(x_3 \mid \mathbf{u_2}, \Sigma_2) \approx 0.011$$

$$\gamma_2(x_3) = \frac{\mathcal{N}(x_3 \mid \mathbf{u_2}, \Sigma_2)}{\mathcal{N}(x_3 \mid \mathbf{u_1}, \Sigma_1) + \mathcal{N}(x_3 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.234$$

Updates:

$$\text{Means:} \quad \mathbf{u_k} = \frac{\sum_i \gamma_k(x_i)\mathbf{x_i}}{\sum_i \gamma_k(x_i)}$$

$$\text{Covariance Matrices:} \quad \Sigma_k = \frac{\sum_i \gamma_k(x_i)(\mathbf{x_i} - \mathbf{u_k})(\mathbf{x_i} - \mathbf{u_k})^T}{\sum_i \gamma_k(x_i)}$$

$$\text{Priors:} \quad \pi_k = \frac{1}{n}\sum_i \gamma_k(x_i)$$

Means:

$$\mathbf{u_1} = \frac{\gamma_1(x_1)\mathbf{x_1} + \gamma_1(x_2)\mathbf{x_2} + \gamma_1(x_3)\mathbf{x_3}}{\gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)}$$

$$= \begin{bmatrix} 2.220 \\ -0.490 \end{bmatrix}$$

$$\mathbf{u_2} = \frac{\gamma_2(x_1)\mathbf{x_1} + \gamma_2(x_2)\mathbf{x_2} + \gamma_2(x_3)\mathbf{x_3}}{\gamma_2(x_1) + \gamma_2(x_2) + \gamma_2(x_3)}$$

$$= \begin{bmatrix} 0.759 \\ 0.867 \end{bmatrix}$$

Covariance Matrices:

$$\Sigma_1 = \frac{\gamma_1(x_1)(\mathbf{x_1} - \mathbf{u_1})(\mathbf{x_1} - \mathbf{u_1})^T + \gamma_1(x_2)(\mathbf{x_2} - \mathbf{u_1})(\mathbf{x_2} - \mathbf{u_1})^T + \gamma_1(x_3)(\mathbf{x_3} - \mathbf{u_1})(\mathbf{x_3} - \mathbf{u_1})^T}{\gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)}$$

$$= \begin{bmatrix} 1.191 & -0.861 \\ -0.861 & 0.728 \end{bmatrix}$$

$$\Sigma_2 = \frac{\gamma_2(x_1)(\mathbf{x_1} - \mathbf{u_2})(\mathbf{x_1} - \mathbf{u_2})^T + \gamma_2(x_2)(\mathbf{x_2} - \mathbf{u_2})(\mathbf{x_2} - \mathbf{u_2})^T + \gamma_2(x_3)(\mathbf{x_3} - \mathbf{u_2})(\mathbf{x_3} - \mathbf{u_2})^T}{\gamma_2(x_1) + \gamma_2(x_2) + \gamma_2(x_3)}$$

$$\Sigma_2 = \begin{pmatrix} 0.954 & -1.044 \\ -1.044 & 1.368 \end{pmatrix}$$

Coefficients:

$$\pi_1 = \frac{1}{3}\left(\gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)\right) \approx 0.393$$

$$\pi_2 = \frac{1}{3}\left(\gamma_2(x_1) + \gamma_2(x_2) + \gamma_2(x_3)\right) \approx 0.607$$

For the 2nd Epoch, using the calculated parameters (again using scipy to help with calculations):

$$\mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1) \approx 0.116, \quad \mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2) \approx 0.149$$

$$\mathcal{N}(x_2 \mid \mathbf{u_1}, \Sigma_1) \approx 0.001, \quad \mathcal{N}(x_2 \mid \mathbf{u_2}, \Sigma_2) \approx 0.207$$

$$\mathcal{N}(x_3 \mid \mathbf{u_1}, \Sigma_1) \approx 0.343, \quad \mathcal{N}(x_3 \mid \mathbf{u_2}, \Sigma_2) \approx 0.012$$

$$\gamma_1(x_1) = \frac{\pi_1 \mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1)}{\pi_1 \mathcal{N}(x_1 \mid \mathbf{u_1}, \Sigma_1) + \pi_2 \mathcal{N}(x_1 \mid \mathbf{u_2}, \Sigma_2)} \approx 0.335$$

$$\gamma_2(x_1) \approx 0.665, \quad \gamma_1(x_2) \approx 0.003, \quad \gamma_2(x_2) \approx 0.997$$

$$\gamma_1(x_3) \approx 0.949, \quad \gamma_2(x_3) \approx 0.051$$

Means:

$$\mathbf{u_1} = \frac{\gamma_1(x_1)\mathbf{x_1} + \gamma_1(x_2)\mathbf{x_2} + \gamma_1(x_3)\mathbf{x_3}}{\gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)} = \begin{bmatrix} 2.472 \\ -0.733 \end{bmatrix}$$

,

$$\mathbf{u_2} = \frac{\gamma_2(x_1)\mathbf{x_1} + \gamma_2(x_2)\mathbf{x_2} + \gamma_2(x_3)\mathbf{x_3}}{\gamma_2(x_1) + \gamma_2(x_2) + \gamma_2(x_3)} = \begin{bmatrix} 0.478 \\ 1.134 \end{bmatrix}$$

Covariances:

$$\Sigma_1 = \frac{\gamma_1(x_1)(\mathbf{x_1} - \mathbf{u_1})(\mathbf{x_1} - \mathbf{u_1})^T + \gamma_1(x_2)(\mathbf{x_2} - \mathbf{u_1})(\mathbf{x_2} - \mathbf{u_1})^T + \gamma_1(x_3)(\mathbf{x_3} - \mathbf{u_1})(\mathbf{x_3} - \mathbf{u_1})^T}{\gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)} = \begin{bmatrix} 0.784 & -0.401 \\ -0.401 & 0.210 \end{bmatrix},$$

$$\Sigma_2 = \frac{\gamma_2(x_1)(\mathbf{x_1}-\mathbf{u_2})(\mathbf{x_1}-\mathbf{u_2})^T + \gamma_2(x_2)(\mathbf{x_2}-\mathbf{u_2})(\mathbf{x_2}-\mathbf{u_2})^T + \gamma_2(x_3)(\mathbf{x_3}-\mathbf{u_2})(\mathbf{x_3}-\mathbf{u_2})^T}{\gamma_2(x_1)+\gamma_2(x_2)+\gamma_2(x_3)} =$$

$$= \begin{bmatrix} 0.428 & -0.631 \\ -0.631 & 1.071 \end{bmatrix}$$

Coefficients:

$$\pi_1 = \frac{1}{3}\left(\gamma_1(x_1)+\gamma_1(x_2)+\gamma_1(x_3)\right) \approx 0.429,$$

$$\pi_2 = \frac{1}{3}\left(\gamma_2(x_1)+\gamma_2(x_2)+\gamma_2(x_3)\right) \approx 0.571$$

2. (a) Taking into account the MAP assumption, we shall look for the normalized joint probabilities calculated previously in the second epoch to check, for each observation, which one is the greatest, for the first cluster or second one, to assign each observation to that cluster.

   Since $\gamma_1(x_1) < \gamma_2(x_1)$, x_1 is assigned to the second cluster.

   $$\gamma_1(x_2) < \gamma_2(x_2) \Rightarrow x_2 \text{ is assigned to the second cluster.}$$

   $$\gamma_1(x_3) > \gamma_2(x_3) \Rightarrow x_3 \text{ is assigned to the first cluster.}$$

   (b) The silhouette of the cluster is the mean of the silhouette of the observations inside that cluster.

   For each point x:

   $$s(x) = 1 - \frac{a(x)}{b(x)}$$

   where: $a(x)$ = average distance of x to the points in its cluster $b(x)$ = minimum distance of x to the points in another cluster

   For the second cluster, which is the one with highest observations in:

   $$s(x_1) = 1 - \frac{a(x_1)}{b(x_1)} = 1 - \frac{d(x_1,x_2)}{d(x_1,x_3)}$$

   $$s(x_2) = 1 - \frac{a(x_2)}{b(x_2)} = 1 - \frac{d(x_2,x_1)}{d(x_2,x_3)}$$

   Euclidean Distance:

   $$d(x_1,x_2) = \sqrt{\sum_i (x_{1i}-x_{2i})^2}$$

   So in our case:

   $$d(x_1,x_2) = \sqrt{(x_{11}-x_{21})^2+(x_{12}-x_{22})^2} = \sqrt{1^2+2^2} = \sqrt{5}$$

$$d(x_1, x_3) = \sqrt{4 + 1} = \sqrt{5}$$

$$d(x_2, x_3) = \sqrt{9 + 1} = \sqrt{10}$$

$$s(x_1) = 1 - \frac{\sqrt{5}}{\sqrt{5}} = 0$$

$$s(x_2) = 1 - \frac{\sqrt{5}}{\sqrt{10}} \approx 0.293$$

$$s(\text{cluster 2}) = \frac{0.293}{2} \approx 0.146$$

Part II: Programming

1. a) Plot of the sum of Squared Errors, using inertia according to the number of clusters, this may allow us to use the elbow method.

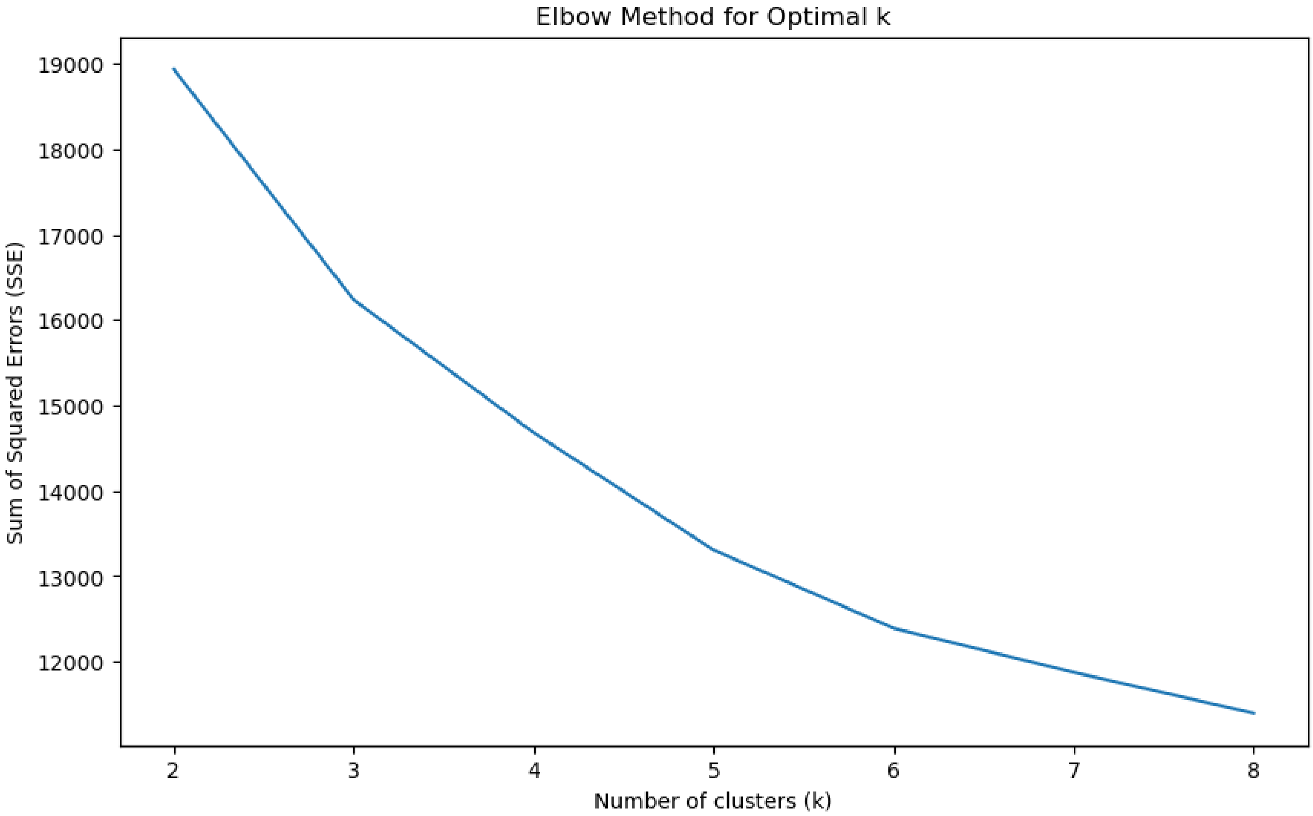

Figure 1: Plot of the Mean Absolute Error for the models indicated.

   b) From the previous graph we need to look out for the elbow in it. The elbow in this case is not too obvious it could be either the Number of clusters = 3 or the number of clusters = 6. Both would be viable and we needed more information to assess which would be better.

   c) k-modes might be better? Yes, K-modes might be a better clustering approach for this dataset because most of the features are categorical, such as 'job', 'marital', 'education', 'default', 'housing', and 'loan'. K-means works better with numerical data, which in this case only represents 2 out of the 8 features selected for analysis. K-modes, on the other hand, is specifically designed to handle categorical variables, making it more suitable for this dataset specifically, where categorical features play a significant role.

2. a) Apply PCA and tell how much variability comes from the top 2 components.

   The variability from the top 2 components corresponds to 23% of the total variability. This usually indicates that 2 components are not enough to be representative of the whole data.

   b) Perform k-means clustering with k=3 and random state = 42 using the original 8 features.

   From the plot before we can't really distinguish the 3 clusters created, this can again be due to the fact that these 2 principal components represents only 23 % of the total variability in the data.

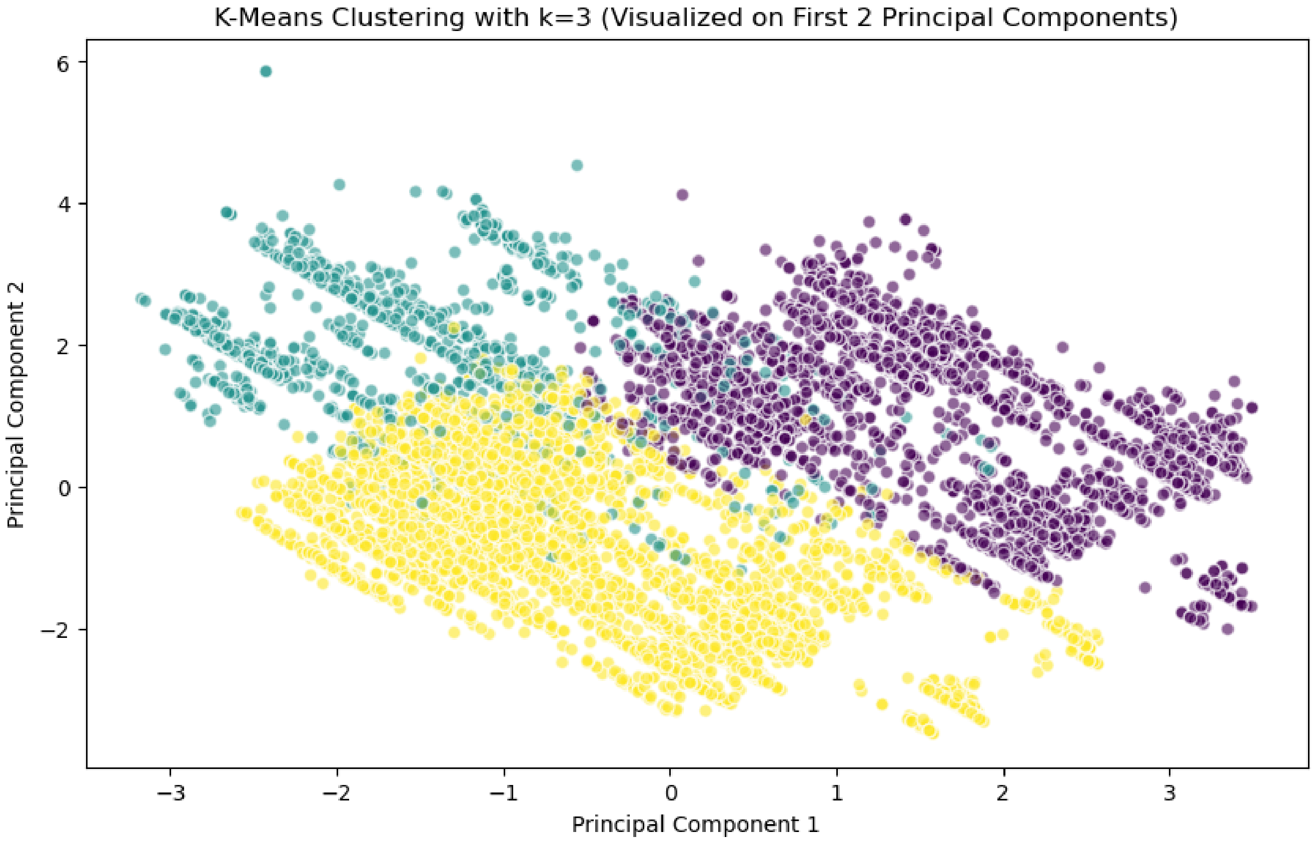

Figure 2: Scatter plot according to the first 2 principal components.

c)

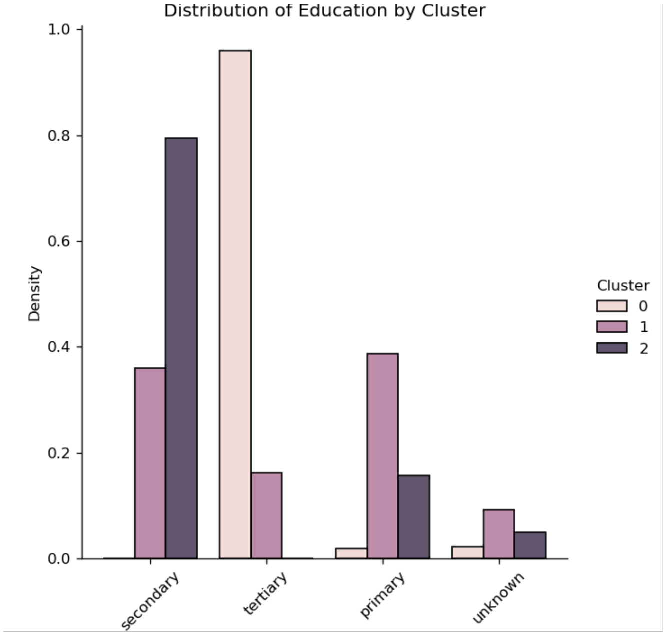

Figure 3: Scatter plot according to the first 2 principal components.

From the education displot we can see that cluster 0 is mostly related to people that have tertiary education, cluster 2 mainly for people that have finished secondary or lower, and the cluster 1 is more evenly distributed along all the 4 classes.

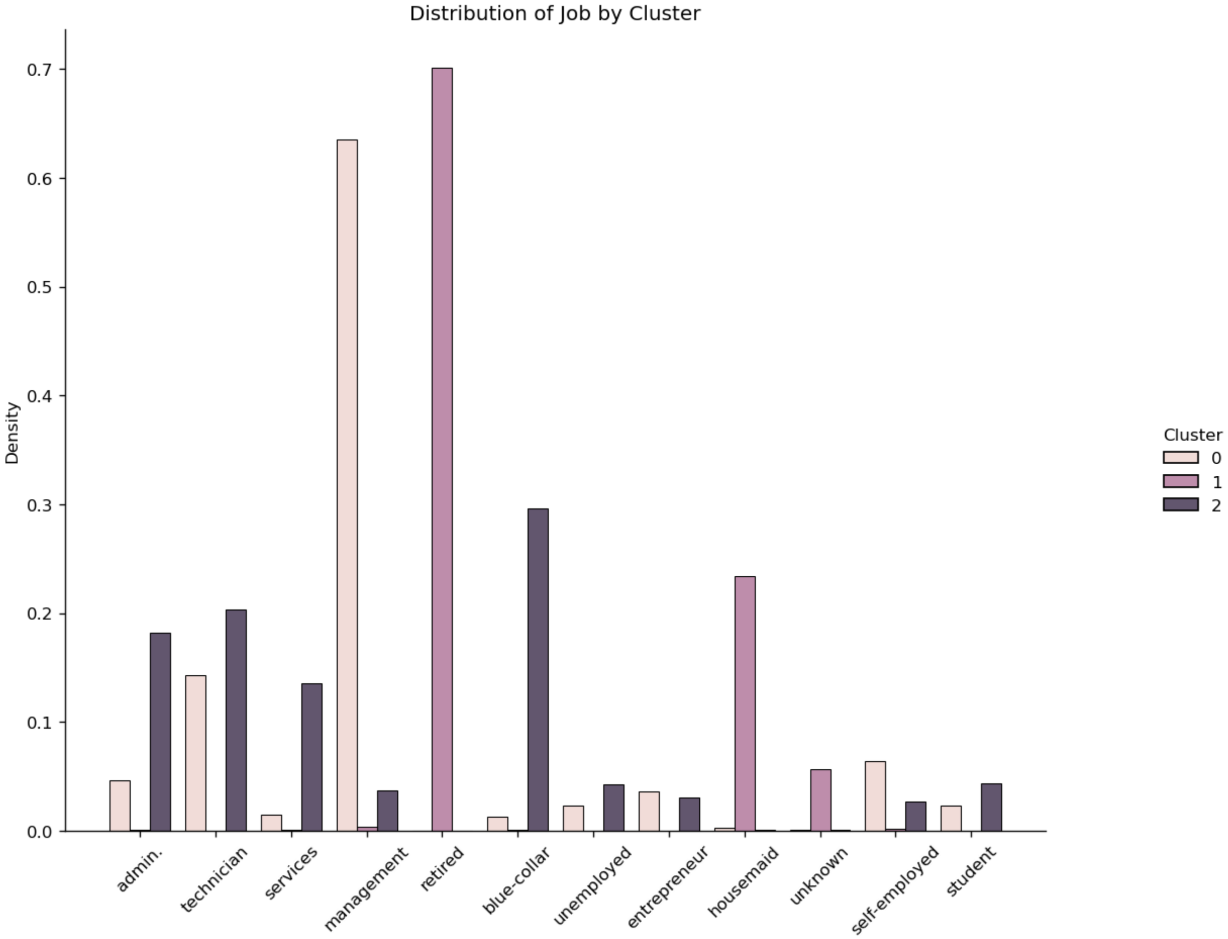

Figure 4: Scatter plot according to the first 2 principal components.

Looking at the jobs displot, we can see that cluster 1 is mostly related to retired people, having another significant portion in housemaids and unknown jobs after that we can see that cluster 0 is mainly related to management and to technicians, and lastly the cluster 3 has a close relationship to the blue-collar, technician, admin and services in a similar proportion.

Looking at both displots we can see that there seems to be a connection between attending tertiary education and having a job in management, and that most of the retired people have probably have not attended tertiary education.

A Code

A.1 Python Script

```python
# %% [markdown]
# # import libraries

# %%
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
import scipy
from scipy import stats
import itertools

# Sklearn
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.metrics import accuracy_score, mean_absolute_error
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# %% [markdown]
# # Read data

# %%
acc_data = pd.read_csv("accounts.csv")

# Remove duplicate rows
acc_data = acc_data.drop_duplicates()

# Remove rows with null values
acc_data = acc_data.dropna()

# Extract the first 8 features
acc_data_inp = acc_data.iloc[:, :8]
acc_data_inp_original = acc_data.iloc[:,:8]
acc_data_tar = acc_data["deposit"]

# Convert categorical columns to numeric using one-hot encoding
acc_data_inp = pd.get_dummies(acc_data_inp, drop_first=True)

# %% [markdown]
# # Basic data visualisation and scale analysis

# %% [markdown]
# ### Target visualisation for curiosity
# %%
plt.hist(acc_data_tar)

# %% [markdown]
# # Input variables visualization for curiosity

# %%
# Plot histograms for numerical variables
numerical_columns = ['age', 'balance']
for column in numerical_columns:
    plt.figure(figsize=(8, 5))
    sns.histplot(data=acc_data_inp_original, x=column, kde=True,
                 multiple='dodge', shrink=0.8)
    plt.title(f'Histogram of {column} by Cluster')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Plot histograms for categorical/binary columns using labels of each person as x-axis
categorical_columns = ['job', 'marital', 'education',
                       'default', 'housing', 'loan']
for column in categorical_columns:
    plt.figure(figsize=(10, 6))
    sns.countplot(data=acc_data_inp_original, x=column, dodge=True)
    plt.title(f'Histogram of {column} by Cluster')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45)
    plt.show()

# %% [markdown]
# # 1) MinMaxScaler Normalization

# %%
scaler = MinMaxScaler()
acc_inp_MM = scaler.fit_transform(acc_data_inp)

# %% [markdown]
# ### a) SSE using _inertia

# %%
SSE =[]
k_values = [2, 3, 4, 5, 6, 7, 8]
for k in k_values:
    kmeans = KMeans(n_clusters=k, max_iter=500, random_state=42)
    kmeans.fit(acc_inp_MM)
    print(kmeans.inertia_)
    SSE.append(kmeans.inertia_)

# %%
# Plot the sum of squared errors (SSE) for different k values
plt.figure(figsize=(10, 6))
plt.plot(k_values, SSE)
plt.xlabel('Number of clusters (k)')
plt.ylabel('Sum of Squared Errors (SSE)')
plt.title('Elbow Method for Optimal k')
plt.show()

# %% [markdown]
# ### b) Comments
# From the previous graph we need to look out for the elbow in it.
# The elbow in this case is not too obvious it could be either the
# Number of clusters = 3 or the number of clusters = 6.

# %% [markdown]
# # c) k-modes might be better?
#  Yes, K-modes might be a better clustering approach for this dataset
#  because most of the features are categorical, such as 'job', 'marital',
#   'education', 'default', 'housing', and 'loan'. K-means works better
#   with numerical data, which in this case only represents 2 out of the
#   8 features selected for analysis. K-modes, on the other hand, is
#   specifically designed to handle categorical variables, making it more
#    suitable for this dataset specifically, where categorical features
#    play a significant role.
#

# %% [markdown]
# # 2) StandardScaler Normalization

# %%
scaler      = StandardScaler()
acc_inp_Std = scaler.fit_transform(acc_data_inp)

# %% [markdown]
# a) Apply PCA and tell how much variability comes from the top 2 components

# %%
# Apply PCA to the data
pca = PCA(n_components=2)
principal_components = pca.fit_transform(acc_inp_Std)
pca_variance = pca.explained_variance_ratio_
# How much variability is explained by the top 2 components?
variability_total        = np.sum(pca_variance)
#percentage_variance_top2 =
print(f"Variability explained by the top 2 components: {variability_total*100:.0f} %")

# %%
# Apply k-means clustering with k=3 and random_state=42 on the original 8 features
kmeans_3 = KMeans(n_clusters=3, random_state=42)
kmeans_3.fit(acc_inp_Std)
```

```
labels = kmeans_3.labels_

# Scatterplot of the first 2 principal components with k-means labels
plt.figure(figsize=(10, 6))
plt.scatter(principal_components[:, 0], principal_components[:, 1],
            c=labels, cmap='viridis', alpha=0.6, edgecolors='w')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('K-Means Clustering with k=3 (Visualized on First 2 Principal Components)')
plt.show()

# %%
clustered_data = acc_data_inp_original.copy()
clustered_data['Cluster'] = labels
# Plot distribution of 'education' according to the clusters
sns.displot(data=clustered_data, x='education', hue='Cluster', multiple="dodge",
            stat='density', shrink=0.8, common_norm=False)
plt.title('Distribution of Education by Cluster')
plt.xlabel('Education')
plt.ylabel('Density')
plt.xticks(rotation=45)
plt.show()


# Plot distribution of 'job' according to the clusters
sns.displot(data=clustered_data, x='job', hue='Cluster', multiple="dodge",
            stat='density', shrink=0.8, common_norm=False)
plt.title('Distribution of Job by Cluster')
plt.xlabel('Job')
plt.ylabel('Density')
plt.xticks(rotation=45)
plt.show()

# %% [markdown]
# From these education displot we can see that cluster 0 is mostly related
#  to people that have tertiary education, cluster 2 mainly for people that
#   have finished secondary or lower, and the cluster 1 is more evenly
#   distributed along all the 4 classes.
# Looking at the jobs displot, we can see that cluster 1 is mostly related
#  to retired people, having another significant portion in housemaids and
#   unknown jobs after that we can see that cluster 0 is mainly related to
#    management and to technicians, and lastly the cluster 3 has a close
#     relationship to the blue-collar, technician, admin and services in a
#     similar proportion.
# Looking at both displots we can see that there seems to be a connection
# between attending tertiary education and having a job in management, and
#  that most of the retired people have probably have not attended
#  tertiary education.
```