Aprendizagem 2023
Homework IV – Group 090
(ist1103335, ist1103573)

# I. Pen-and-paper [11v]

Given the following observations,

$$\{\begin{pmatrix}1\\0.6\\0.1\end{pmatrix},\begin{pmatrix}0\\-0.4\\0.8\end{pmatrix},\begin{pmatrix}0\\0.2\\0.5\end{pmatrix},\begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\}.$$

Consider a Bayesian clustering that assumes $\{y_1\} \perp \{y_2, y_3\}$, two clusters following a Bernoulli distribution on $y_1$ ($p_1$ and $p_2$), a multivariate Gaussian on $\{y_2, y_3\}$ ($N_1$ and $N_2$), and the following initial mixture:

$$\begin{aligned}\pi_1 &= 0.5, \pi_2 = 0.5\\ p_1 &= P(y_1 = 1) = 0.3, p_2 = P(y_1 = 1) = 0.7\\ \mathcal{N}_1(\boldsymbol{\mu}_1 &= \begin{pmatrix}1\\1\end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix})\\ \mathcal{N}_2(\boldsymbol{\mu}_2 &= \begin{pmatrix}0\\0\end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{pmatrix}1.5 & 1\\1 & 1.5\end{pmatrix})\end{aligned}$$

Note: you can solve this exercise by neglecting $y_1$ and still scoring up to 70

### 1) [6v] Perform one epoch of the EM clustering algorithm and determine the new parameters.

Hint: we suggest you to use numpy and scipy, however disclose the intermediary results step by step.

To perform one epoch of the EM clustering algorithm, we need to update the priors $\pi_1$, $\pi_2$, the means $\boldsymbol{\mu}_1$ $\boldsymbol{\mu}_2$, the covariance matrices $\Sigma_1$ and $\Sigma_2$ and the probabilities $p_1$ and $p_2$

**Expectation Step:** For this we compute the probability P($c_k|x_i$) for each observation $x_i$ e cluster $c_k$:

$$\gamma_{ki} = P\left(c_k|x_i\right) = \frac{P\left(x_i|c_k\right)\cdot P\left(c_k\right)}{P\left(x_i\right)} = \frac{\pi_k \cdot \mathcal{N}_k\left(x_{i[y_2,y_3]}|\mu_k, \Sigma_k\right)\cdot Bernoulli\left(x_{i[y_1]}|p_k\right)}{\sum\limits_k\left(\pi_k \cdot \mathcal{N}_k\left(x_{i[y_2,y_3]}|\mu_k, \Sigma_k\right)\cdot Bernoulli\left(x_{i[y_1]}|p_k\right)\right)}$$

As P($x_i$) is invariant across components, we can simply calculate:

$$P\left(c_k, x_i\right) = P\left(x_i|c_k\right)\cdot P\left(c_k\right) = \pi_k \cdot \mathcal{N}_k\left(x_{i[y_2,y_3]}|\mu_k, \Sigma_k\right)\cdot Bernoulli\left(x_{i[y_1]}|p_k\right)$$

and then we **normalize** it:

$$\gamma_{ki} = P\left(c_k|x_i\right) = \frac{P\left(c_k, x_i\right)}{\Sigma_j P\left(c_j, x_i\right)}$$

Then assuming $x_1 = \begin{pmatrix} 1 \\ 0.6 \\ 0.1 \end{pmatrix}, x_2 = \begin{pmatrix} 0 \\ -0.4 \\ 0.8 \end{pmatrix}, x_3 = \begin{pmatrix} 0 \\ 0.2 \\ 0.5 \end{pmatrix}, x_4 = \begin{pmatrix} 1 \\ 0.4 \\ -0.1 \end{pmatrix}$:

$$\begin{aligned} P\left(c_1, x_1\right) &= \pi_1 \cdot \mathcal{N}_1 \left(x_{1[y_2,y_3]} | \mu_1, \Sigma_1\right) \cdot \mathit{Bernoulli}\left(x_{1[y_1]} | p_1\right) \\ &= 0.5 \cdot 0.06658 \cdot 0.3 \\ &= 0.009986 \end{aligned}$$

$$P\left(c_1, x_1\right) = 0.009986,\ P\left(c_1, x_2\right) = 0.017517,\ P\left(c_1, x_3\right) = 0.023931, P\left(c_1, x_4\right) = 0.008857$$

$$P\left(c_2, x_1\right) = 0.041866,\ P\left(c_2, x_2\right) = 0.010228,\ P\left(c_2, x_3\right) = 0.019437, P\left(c_2, x_4\right) = 0.043575$$

Then we normalize the values to compute the $\gamma_{ik}$:

$$\begin{aligned} \gamma_{11} &= \frac{P\left(c_1, x_1\right)}{P\left(c_1, x_1\right) + P\left(c_2, x_1\right)} \\ &= \frac{0.009986}{0.009986 + 0.041866} \\ &= 0.192590 \end{aligned}$$

$$\gamma_{11} = 0.192590,\ \gamma_{21} = 0.631345,\ \gamma_{31} = 0.551811,\ \gamma_{41} = 0.168924$$

$$\gamma_{12} = 0.807410,\ \gamma_{22} = 0.368655,\ \gamma_{32} = 0.448189,\ \gamma_{42} = 0.831076$$

Where $\gamma_{21}$ represents the probability of the observation 2 belonging to cluster 1.

**Maximization Step:** Update the parameters for each cluster k:

$$\begin{aligned} N_k &= \sum_{\eta=1}^{N} \gamma_{\eta k} \\ \mu_k &= \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma_{\eta k} \cdot \mathbf{x}_{\eta[y_2,y_3]} \\ \Sigma_k &= \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma_{\eta k} \cdot \left(\mathbf{x}_{\eta[y_2,y_3]} - \mu_k\right) \cdot \left(\mathbf{x}_{\eta[y_2,y_3]} - \mu_k\right)^T \\ p_k &= \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma_{\eta k} \cdot \mathbf{x}_{\eta[y_1]} \\ \pi_k &= p\left(c_k = 1\right) = \frac{N_k}{N} \end{aligned}$$

Now we calculate the updated parameters using these formulas:

- $\mathbf{N_k}$ :

$$\begin{aligned} N_1 &= \sum_{i=1}^{4} \gamma_{\eta 1} = 0.192590 + 0.631345 + 0.551811 + 0.168924 \\ &= \mathbf{1.544670} \\ N_2 &= \mathbf{2.455330} \end{aligned}$$

- $\mu_{\mathbf{k}}$ :

$$\begin{aligned}
\mu_1' &= \frac{1}{N_1} \cdot \sum_{\eta=1}^{4} \gamma_{\eta 1} \cdot \mathbf{x}_{\eta[y_2,y_3]} \\
&= \frac{0.192590 \cdot \begin{bmatrix} 0.6 \\ 0.1 \end{bmatrix} + 0.631345 \cdot \begin{bmatrix} -0.4 \\ 0.8 \end{bmatrix} + 0.551811 \cdot \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix} + 0.168924 \cdot \begin{bmatrix} 0.4 \\ -0.1 \end{bmatrix}}{1.544670} \\
&= \begin{bmatrix} 0.026509 \\ 0.507130 \end{bmatrix} \\
\mu_2' &= \begin{bmatrix} 0.309145 \\ 0.210420 \end{bmatrix}
\end{aligned}$$

- $\mathbf{\Sigma_k}$ :

$$\begin{aligned}
\Sigma_1' &= \frac{1}{N_1} \cdot \sum_{\eta=1}^{4} \gamma_{\eta 1} \cdot \left(\mathbf{x}_{\eta[y_2,y_3]} - \mu_1\right) \cdot \left(\mathbf{x}_{\eta[y_2,y_3]} - \mu_1\right)^T \\
&= \Bigg(0.192590 \cdot \begin{bmatrix} 0.573491 \\ -0.40713 \end{bmatrix} \begin{bmatrix} 0.573491 \\ -0.40713 \end{bmatrix}^T + 0.631345 \cdot \begin{bmatrix} -0.426509 \\ 0.29287 \end{bmatrix} \begin{bmatrix} -0.426509 \\ 0.29287 \end{bmatrix}^T \\
&\quad +0.551811 \cdot \begin{bmatrix} 0.173491 \\ -0.00713 \end{bmatrix} \begin{bmatrix} 0.173491 \\ -0.00713 \end{bmatrix}^T + 0.168924 \cdot \begin{bmatrix} 0.375938 \\ -0.60713 \end{bmatrix} \begin{bmatrix} 0.375938 \\ -0.60713 \end{bmatrix}^T \Bigg) \\
&= \begin{bmatrix} 0.141365 & -0.105405 \\ -0.105405 & 0.096053 \end{bmatrix} \\
\Sigma_2' &= \begin{bmatrix} 0.108293 & -0.088652 \\ -0.088652 & 0.104123 \end{bmatrix}
\end{aligned} \tag{1}$$

- $\mathbf{p_k}$ :

$$\begin{aligned}
p_1' &= \frac{1}{N_1} \cdot \sum_{\eta=1}^{4} \gamma_{\eta 1} \cdot \mathbf{x}_{\eta[y_1]} \\
&= \frac{0.192590 \cdot \begin{bmatrix} 1 \end{bmatrix} + 0.631345 \cdot \begin{bmatrix} 0 \end{bmatrix} + 0.551811 \cdot \begin{bmatrix} 0 \end{bmatrix} + 0.168924 \cdot \begin{bmatrix} 1 \end{bmatrix}}{1.544670} \\
&= \mathbf{0.234039} \\
p_2' &= \mathbf{0.667318}
\end{aligned}$$

- $\pi_{\mathbf{k}}$ :

$$\begin{aligned}
\pi_1' &= \frac{N_1}{N_1 + N_2} = \frac{1.544670}{1.544670 + 2.455330} \\
&= \mathbf{0.386168} \\
\pi_2' &= \mathbf{0.613832}
\end{aligned}$$

**2) [2v] Given the new observation, $\boldsymbol{x}_{\mathbf{new}} = \begin{pmatrix} 1 \\ 0.3 \\ 0.7 \end{pmatrix}$, determine the cluster memberships (posteriors).**

$$\begin{aligned} P\left(c_1, x_{new}\right) &= \pi_1 \cdot \mathcal{N}_1\left(x_{new[y_2,y_3]} | \mu_1, \Sigma_1\right) \cdot Bernoulli\left(x_{new[y_1]} | p_1\right) \\ &= 0.386168 \cdot 0.027076 \cdot 0.234039 \\ &= 0.080290 \\ P\left(c_2, x_{new}\right) &= 0.919710 \end{aligned}$$

The posteriors are then calculated by:

$$\begin{aligned} \gamma_{new1} &= \frac{P\left(c_1, x_{new}\right)}{P\left(c_1, x_{new}\right) + P\left(c_2, x_{new}\right)} \\ &= \frac{0.080290}{0.080290 + 0.919710} \\ &= \mathbf{0.002447} \\ \gamma_{new2} &= \mathbf{0.028031} \end{aligned}$$

**3) [2.5v] Performing a hard assignment of observations to clusters under a ML assumption, identify the silhouette for the both clusters under a Manhattan distance.**

The observations are:

$$x_1 = \begin{pmatrix} 1 \\ 0.6 \\ 0.1 \end{pmatrix},\ x_2 = \begin{pmatrix} 0 \\ -0.4 \\ 0.8 \end{pmatrix},\ x_3 = \begin{pmatrix} 0 \\ 0.2 \\ 0.5 \end{pmatrix},\ x_4 = \begin{pmatrix} 1 \\ 0.4 \\ -0.1 \end{pmatrix}$$

The likelihood for each $x_i$ is determined by:

$$P\left(x_i | c_k\right) = \mathcal{N}_1\left(x_{i[y_2,y_3]} | \mu_k, \Sigma_k\right) \cdot Bernoulli\left(x_{new[i]} | p_k\right)$$

So we calculate the likelihoods:

1. For $x_1$ in $c_1$: 0.23147434
   For $x_1$ in $c_2$: 0.94954252
   So $x_1$ belongs to $c_2$

2. For $x_2$ in $c_1$: 1.26633248
   For $x_2$ in $c_2$: 0.08873672
   So $x_2$ belongs to $c_1$

3. For $x_3$ in $c_1$: 1.4381104
   For $x_3$ in $c_2$: 0.4541745
   So $x_3$ belongs to $c_1$

4. For $x_4$ in $c_1$: 0.02076523
   For $x_4$ in $c_2$: 0.72331198
   So $x_4$ belongs to $c_2$

Therefore we have the clusters:

$$c_1 = \{x_2, x_3\}\ and\ c_2 = \{x_1, x_4\}$$

Preserving the Manhattan distance assumption, let us compute the silhouette of $c_1$:

The silhouette is $s_i = \frac{b-a}{max\{a,b\}}$

- a is the average distance of point $i$ to the others in same cluster
  $a(i) = \frac{1}{|C_i|-1} \sum_{\substack{j \in C_i \\ j \neq i}} d(i,j)$
- b is the minimum of the average distances of point $i$ to the points in each other cluster
  $b(i) = \min_{j \neq i} \left( \frac{1}{|C_j|} \sum_{k \in C_j} d(i,k) \right)$

First, we calculate a and b for $x_2$:

$$a = |0-0| + |-0.4-0.2| + |0.8-0.5| = 0.9$$

$$b = \frac{|0-1| + |-0.4-0.6| + |0.8-0.1| + |0-1| + |-0.4-0.4| + |0.8+0.1|}{2} = 2.7$$

Therefore the silhouette for $x_2$ is $s(x_2) = \frac{2.7-0.9}{2.7} = 0,6(6)$
Then we do the same for the $x_3$ and we obtain a silhouette of 0.50.

Therefore the average silhouette for cluster 1 is $s(c_1) =$ **0.58 (3)**.
We do the same steps for $c_2$ and we determined that the average silhouette for cluster 2 is $s(c_2) =$ **0.8(2)**

**4) [0.5v] Knowing the purity of the clustering solution is 0.75, identify the number of possible classes (ground truth).**

Purity is an external measure that assesses how many of the clusters contain only a single class or label. The formula for purity is:

**purity(C, L)** $= \frac{1}{N}\sum_{k=1}^{K} \max_j (|c_k \cap l_j|)$

Where,

- $N$ is the total number of observations
- $K$ is the number of clusters
- $C_k$ is the set of points in cluster
- $L_j$ is the set of points in the true class $j$

We want to identify the number of possible classes for a purity of 0.75 knowing from the previous question that $N$ is 4, $K$ is 2 and we have $c_1$ and $c_2$:

$$c_1 = \{\begin{pmatrix}0\\-0.4\\0.8\end{pmatrix}, \begin{pmatrix}0\\0.2\\0.5\end{pmatrix}\},\ c_2 = \{\begin{pmatrix}1\\0.6\\0.1\end{pmatrix}, \begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\}$$

For this value of purity we have 2 possibilities. The first **possible number of classes is 2**. Let's imagine we have:

$$l_1 = \{\begin{pmatrix}0\\-0.4\\0.8\end{pmatrix}, \begin{pmatrix}0\\0.2\\0.5\end{pmatrix}, \begin{pmatrix}1\\0.6\\0.1\end{pmatrix}\} \text{ and } l_2 = \{\begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\}$$

Therefore:

$$\begin{aligned} purity(C, L) &= \frac{1}{4}\left(\max\{|c_1 \cap l_1|, |c_1 \cap l_2|\} + \max\{|c_2 \cap l_1|, |c_2 \cap l_2|\}\right) \\ &= \frac{1}{4}\left(\max\{2, 0\} + \max\{1, 1\}\right) = \frac{1}{4}(2+1) \\ &= \frac{1}{4}(3) = \frac{3}{4} = 0.75 \end{aligned}$$

The other **possible number of classes is 3**. Let's imagine we have:

$$l_1 = \{\begin{pmatrix}0\\-0.4\\0.8\end{pmatrix}, \begin{pmatrix}0\\0.2\\0.5\end{pmatrix}\},\ l_2 = \{\begin{pmatrix}1\\0.4\\-0.1\end{pmatrix}\} \text{ and } l_3 = \{\begin{pmatrix}1\\0.6\\0.1\end{pmatrix}\}$$

Therefore:

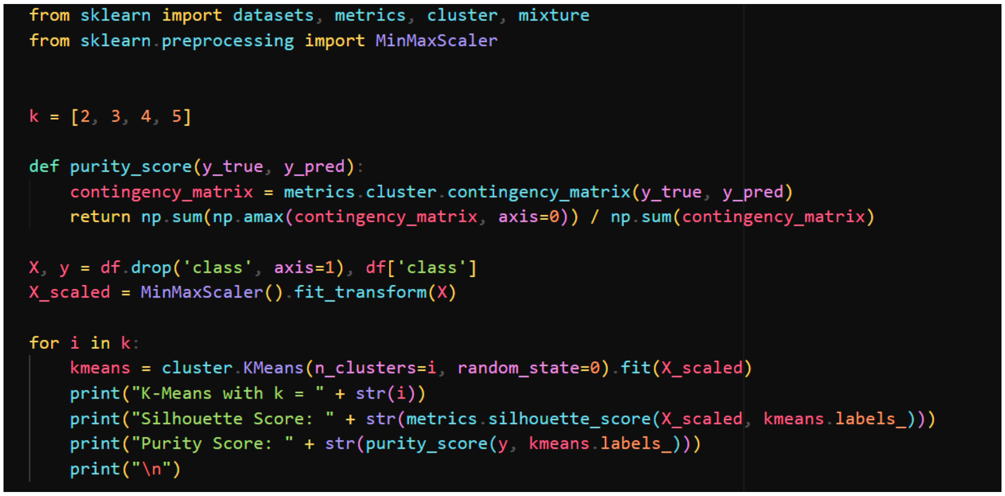
$$\begin{aligned} purity(C, L) &= \frac{1}{4}\left(\max\{|c_1 \cap l_1|, |c_1 \cap l_2|, |c_1 \cap l_3|\} + \max\{|c_2 \cap l_1|, |c_2 \cap l_2|, |c_2 \cap l_3|\}\right) \\ &= \frac{1}{4}\left(\max\{2, 0, 0\} + \max\{0, 1, 1\}\right) = \frac{1}{4}(2+1) \\ &= \frac{1}{4}(3) = \frac{3}{4} = 0.75 \end{aligned}$$

We cant have only 1 class because the Purity would be 1 and we can't have 4 classes because the purity would be 0.5.

## II. Programming and critical analysis [9v]

Recall the column_diagnosis.arff dataset from previous homeworks. For the following exercises, normalize the data using sklearn's MinMaxScaler.

**1) [4v] Using sklearn, apply k-means clustering fully unsupervisedly on the normalized data with $k \in \{2, 3, 4, 5\}$ (random=0 and remaining parameters as default). Assess the silhouette and purity of the produced solutions.**



```
from sklearn import datasets, metrics, cluster, mixture
from sklearn.preprocessing import MinMaxScaler


k = [2, 3, 4, 5]

def purity_score(y_true, y_pred):
    contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
    return np.sum(np.amax(contingency_matrix, axis=0)) / np.sum(contingency_matrix)

X, y = df.drop('class', axis=1), df['class']
X_scaled = MinMaxScaler().fit_transform(X)

for i in k:
    kmeans = cluster.KMeans(n_clusters=i, random_state=0).fit(X_scaled)
    print("K-Means with k = " + str(i))
    print("Silhouette Score: " + str(metrics.silhouette_score(X_scaled, kmeans.labels_)))
    print("Purity Score: " + str(purity_score(y, kmeans.labels_)))
    print("\n")
```

**K-Means Clustering with k = 2**

Silhouette Score: 0.36044124340441114 Purity Score: 0.632258064516129

**K-Means Clustering with k = 3**

Silhouette Score: 0.29579055730002257 Purity Score: 0.667741935483871

**K-Means Clustering with k = 4**

Silhouette Score: 0.27442402122340176 Purity Score: 0.6612903225806451

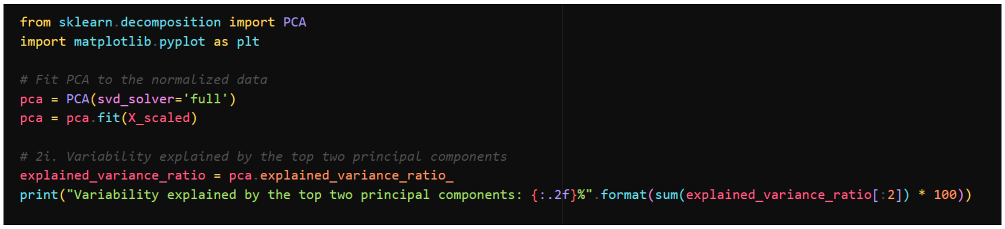
**K-Means Clustering with k = 5**

Silhouette Score: 0.23823928397844843 Purity Score: 0.6774193548387096

**2) [2v] Consider the application of PCA after the data normalization:**

1. **Identify the variability explained by the top two principal components.**
2. **For each one of these two components, sort the input variables by relevance by inspecting the absolute weights of the linear projection.**

i.



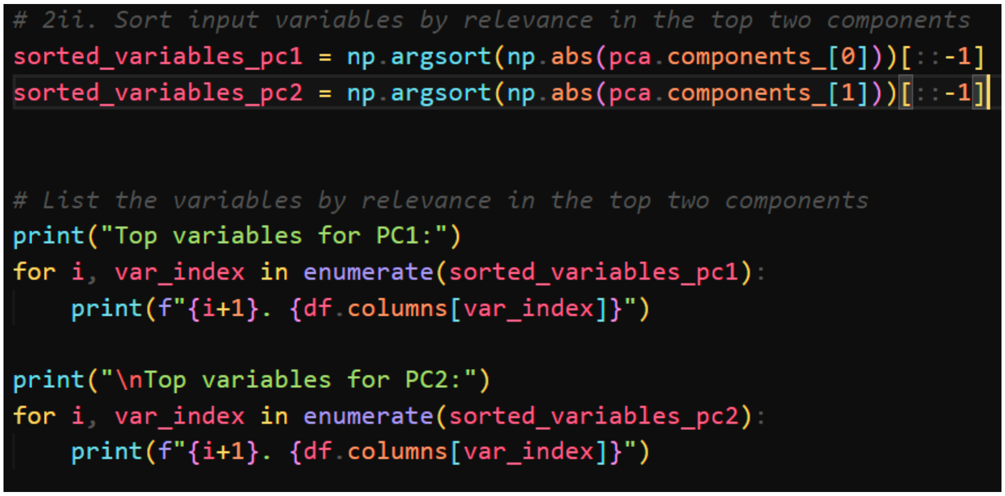
```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Fit PCA to the normalized data
pca = PCA(svd_solver='full')
pca = pca.fit(X_scaled)

# 2i. Variability explained by the top two principal components
explained_variance_ratio = pca.explained_variance_ratio_
print("Variability explained by the top two principal components: {:.2f}%".format(sum(explained_variance_ratio[:2]) * 100))
```

**Variability explained by the top two principal components**: 77.14%

ii.



```
# 2ii. Sort input variables by relevance in the top two components
sorted_variables_pc1 = np.argsort(np.abs(pca.components_[0]))[::-1]
sorted_variables_pc2 = np.argsort(np.abs(pca.components_[1]))[::-1]


# List the variables by relevance in the top two components
print("Top variables for PC1:")
for i, var_index in enumerate(sorted_variables_pc1):
    print(f"{i+1}. {df.columns[var_index]}")

print("\nTop variables for PC2:")
for i, var_index in enumerate(sorted_variables_pc2):
    print(f"{i+1}. {df.columns[var_index]}")
```

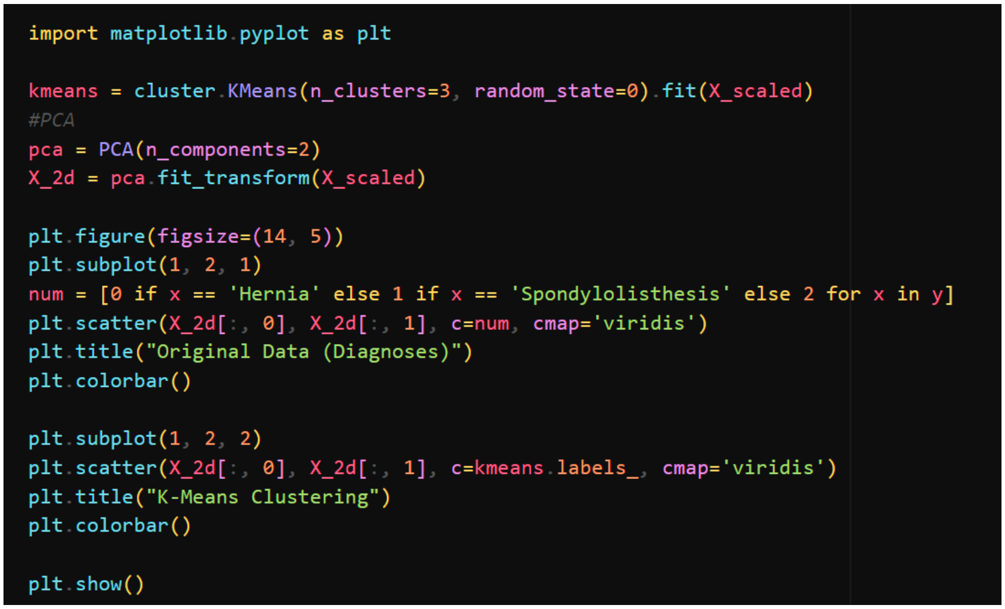
**Top variables for PC1:**

1. pelvic_incidence
2. lumbar_lordosis_angle
3. pelvic_tilt
4. sacral_slope
5. degree_spondylolisthesis
6. pelvic_radius

**Top variables for PC2:**

1. pelvic_tilt
2. pelvic_radius
3. sacral_slope
4. pelvic_incidence
5. lumbar_lordosi0s_angle
6. degree_spondylolisthesis

**3) [2v] Visualize side-by-side the data using: i) the ground diagnoses, and ii) the previously learned $k = 3$ clustering solution. To this end, project the normalized data onto a 2-dimensional data space using PCA and then color observations using the reference and cluster annotations.**



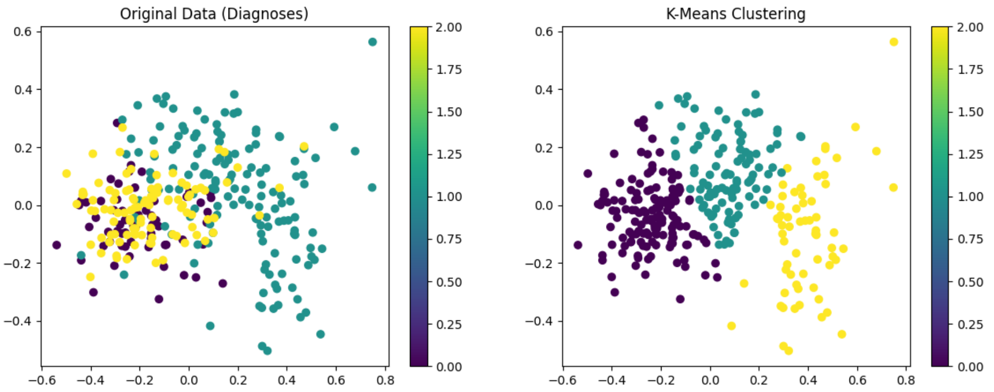
```
import matplotlib.pyplot as plt

kmeans = cluster.KMeans(n_clusters=3, random_state=0).fit(X_scaled)
#PCA
pca = PCA(n_components=2)
X_2d = pca.fit_transform(X_scaled)

plt.figure(figsize=(14, 5))
plt.subplot(1, 2, 1)
num = [0 if x == 'Hernia' else 1 if x == 'Spondylolisthesis' else 2 for x in y]
plt.scatter(X_2d[:, 0], X_2d[:, 1], c=num, cmap='viridis')
plt.title("Original Data (Diagnoses)")
plt.colorbar()

plt.subplot(1, 2, 2)
plt.scatter(X_2d[:, 0], X_2d[:, 1], c=kmeans.labels_, cmap='viridis')
plt.title("K-Means Clustering")
plt.colorbar()

plt.show()
```



**4) [1v] Considering the results from questions (1) and (3), identify two ways on how clustering can be used to characterize the population of ill and healthy individuals.**

Clustering can be used to characterize the population of ill and healthy individuals in two ways:

**Identifying Subgroups**: Clustering helps identify different subgroups within the ill or healthy populations based on common characteristics. This can help identify the characteristics that are most closely associated with the illness.

**Risk Assessment**: Clustering can also be used to assess the risk of developing an illness. For example, if a person is in a cluster with many ill people, they may be at higher risk of developing the illness than someone in a cluster with few ill people.