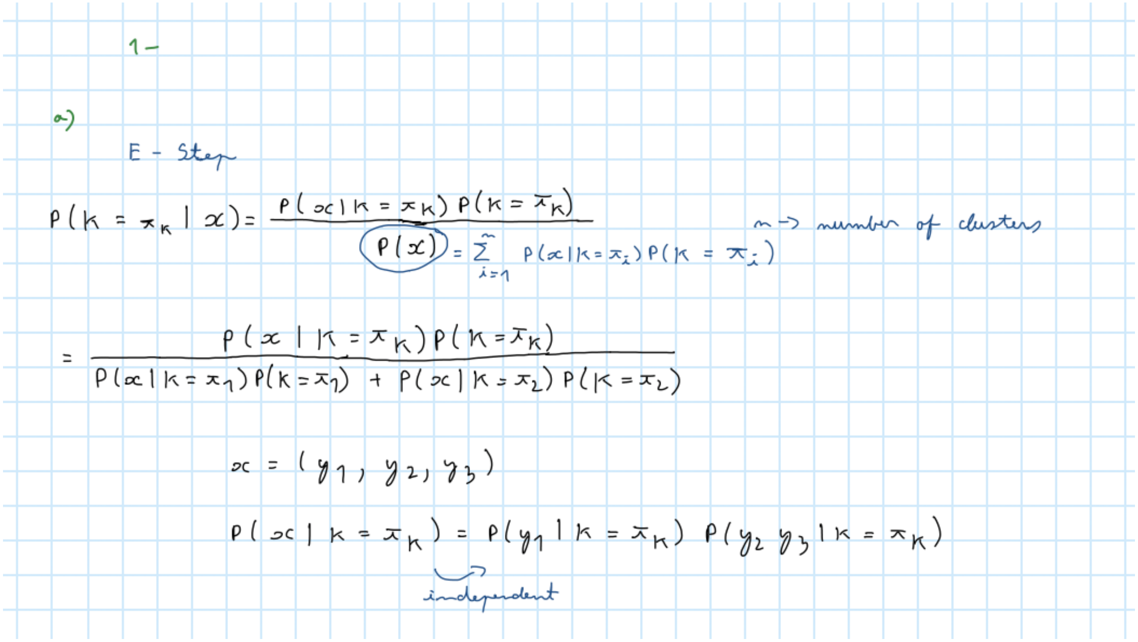
Aprendizagem 2023
Homework IV – Group 023
(ist102848, ist103525)

Part I: Pen and paper

1. Perform one epoch of the EM clustering algorithm and determine the new parameters.

EM algorithm is composed by two steps, E-STEP and M-STEP, in the E-STEP we calculate the posteriors for every observation and cluster. To do this we use the following formulas:



1–

a)

E - Step

$$P(K = \pi_K \mid x) = \frac{P(x \mid K = \pi_K)\,P(K = \pi_K)}{\boxed{P(x)}} \qquad \boxed{P(x)} = \sum_{i=1}^{m} P(x \mid K = \pi_i)\,P(K = \pi_i)$$

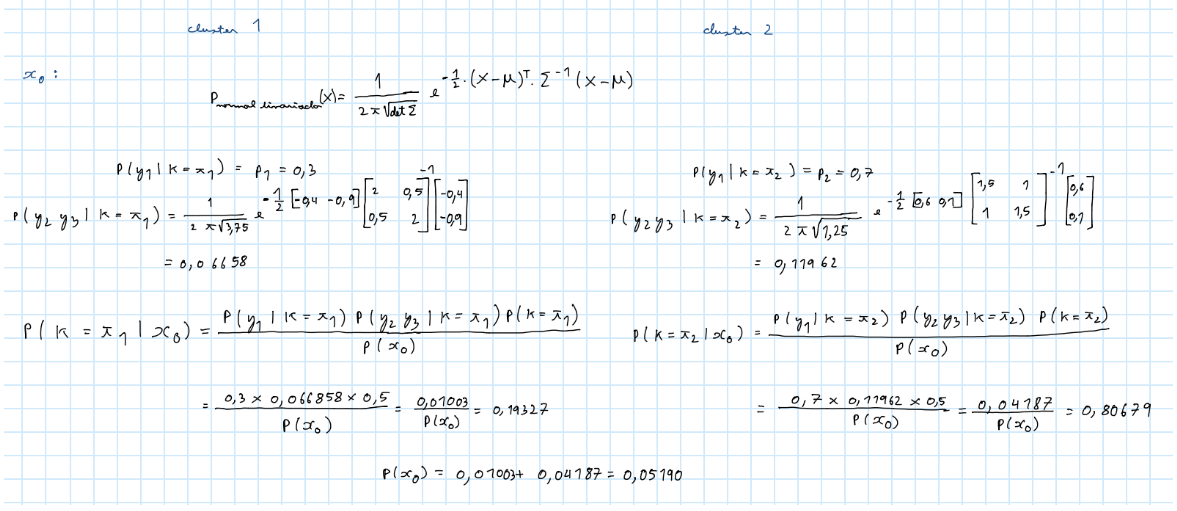
$m \to$ number of clusters

$$= \frac{P(x \mid K = \pi_K)\,P(K = \pi_K)}{P(x \mid K = \pi_1)\,P(K = \pi_1) + P(x \mid K = \pi_2)\,P(K = \pi_2)}$$

$$x = (y_1, y_2, y_3)$$

$$P(x \mid K = \pi_K) = \underbrace{P(y_1 \mid K = \pi_K)\;P(y_2\, y_3 \mid K = \pi_K)}_{\text{independent}}$$

To calculate the posterior, we begin by doing some auxiliary calculations, computing the probability of 'X' occurring in both the Bernoulli and Bivariate Normal distributions for each cluster.



cluster 1 | cluster 2

x_0 :

$$P_{\text{normal bivariada}}(X) = \frac{1}{2\pi\sqrt{\det \Sigma}}\, e^{-\frac{1}{2}\cdot(X-\mu)^T \cdot \Sigma^{-1}(X-\mu)}$$

$$P(y_1 \mid K = \pi_1) = p_1 = 0{,}3$$

$$P(y_2\, y_3 \mid K = \pi_1) = \frac{1}{2\pi\sqrt{3{,}75}}\, e^{-\frac{1}{2}\begin{bmatrix}-0{,}4 & -0{,}9\end{bmatrix}\begin{bmatrix}2 & 0{,}5\\ 0{,}5 & 2\end{bmatrix}^{-1}\begin{bmatrix}-0{,}4\\ -0{,}9\end{bmatrix}}$$

$$= 0{,}06658$$

$$P(y_1 \mid K = \pi_2) = p_2 = 0{,}7$$

$$P(y_2\, y_3 \mid K = \pi_2) = \frac{1}{2\pi\sqrt{1{,}25}}\, e^{-\frac{1}{2}\begin{bmatrix}0{,}6 & 0{,}1\end{bmatrix}\begin{bmatrix}1{,}5 & 1\\ 1 & 1{,}5\end{bmatrix}^{-1}\begin{bmatrix}0{,}6\\ 0{,}1\end{bmatrix}}$$

$$= 0{,}11962$$

$$P(K = \pi_1 \mid x_0) = \frac{P(y_1 \mid K = \pi_1)\,P(y_2\, y_3 \mid K = \pi_1)\,P(K = \pi_1)}{P(x_0)}$$

$$= \frac{0{,}3 \times 0{,}066858 \times 0{,}5}{P(x_0)} = \frac{0{,}01003}{P(x_0)} = 0{,}19327$$

$$P(K = \pi_2 \mid x_0) = \frac{P(y_1 \mid K = \pi_2)\,P(y_2\, y_3 \mid K = \pi_2)\,P(K = \pi_2)}{P(x_0)}$$

$$= \frac{0{,}7 \times 0{,}11962 \times 0{,}5}{P(x_0)} = \frac{0{,}04187}{P(x_0)} = 0{,}80679$$

$$P(x_0) = 0{,}01003 + 0{,}04187 = 0{,}05190$$

We continue doing the same process for the remaining posteriors.

	cluster 1	cluster 2
x_1:	$p(k = \pi_1 \mid x_1) = 0,63135$	$p(k = \pi_2 \mid x_1) = 0,36865$
x_2:	$p(k = \pi_1 \mid x_2) = 0,55181$	$p(k = \pi_2 \mid x_2) = 0,44819$
x_3:	$p(k = \pi_1 \mid x_3) = 0,16892$	$p(k = \pi_2 \mid x_3) = 0,83108$

After the E-STEP is completed, we start updating the priors and the new parameters (M-STEP). First we calculate the N_1 and N_2 for future calculations.

M-STEP

• calculate N_K

$$N_K = \sum_{i=0}^{n_0} p(k = \pi_K \mid x_i)$$

$n_0 \rightarrow$ number of observations - 1

$n_K \rightarrow$ number of clusters

$$N = \sum_{K=1}^{n_K} N_K$$

$$N_1 = 0,19259 + 0,63735 + 0,55187 + 0,16892 = 1,54467$$

$$N_2 = 2,45533$$

$$N = 1,54467 + 2,45533 = 4,0$$

Now we update the priors using the following formula:

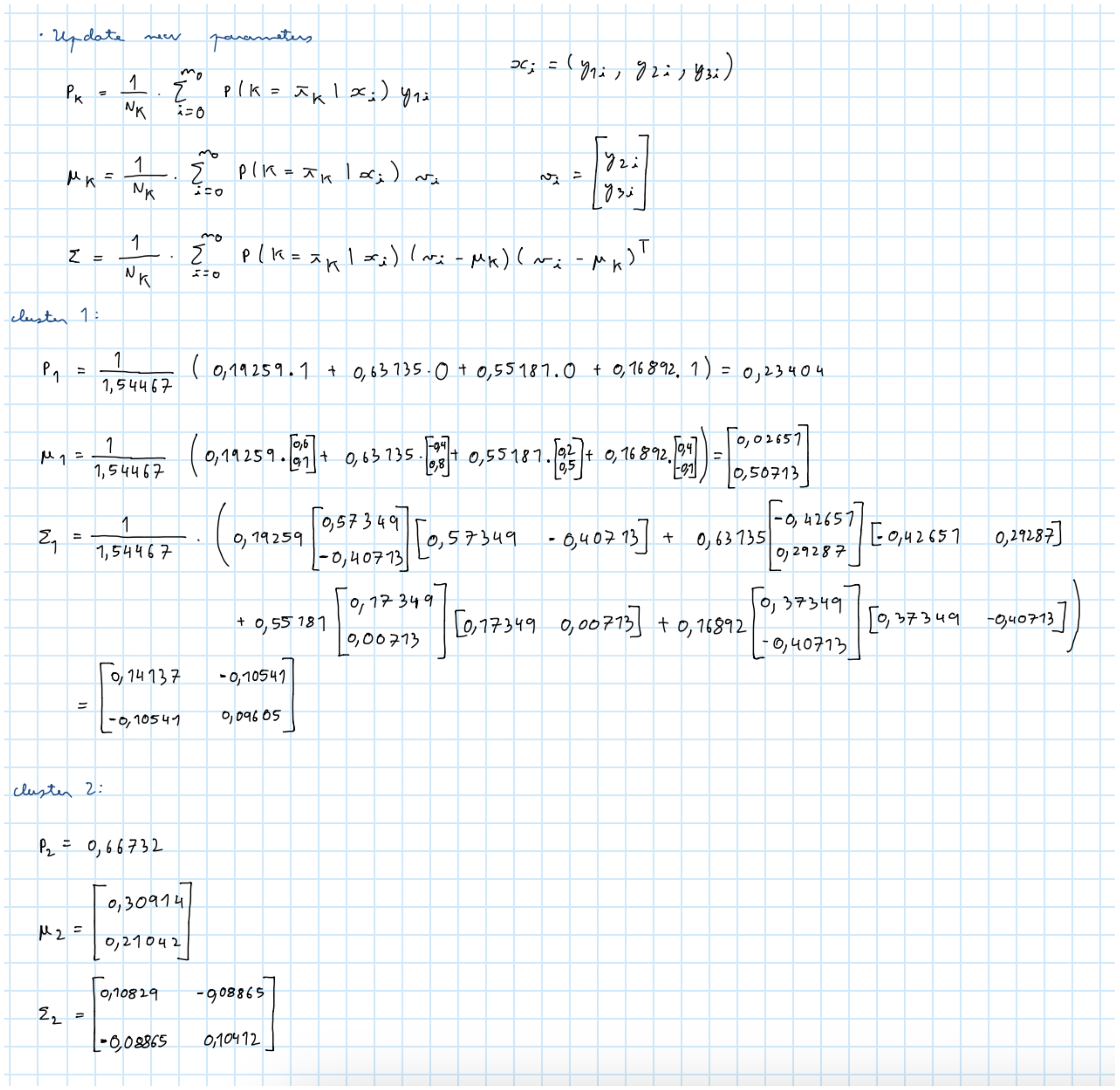
• Update priors

$$\text{prior} = p(k = \pi_K) = \frac{N_K}{N}$$

$$p(k = \pi_1) = \frac{N_1}{N} = \frac{1,54467}{4,0} = 0,38617$$

$$p(k = \pi_2) = \frac{N_2}{N} = 0,61383$$

And finally, to complete the M-STEP, we update the parameters of both distributions.



· Update new parameters

$$x_i = (y_{1i}, y_{2i}, y_{3i})$$

$$P_K = \frac{1}{N_K} \cdot \sum_{i=0}^{m_0} p(K = \pi_K \mid x_i)\, y_{1i}$$

$$\mu_K = \frac{1}{N_K} \cdot \sum_{i=0}^{m_0} p(K = \pi_K \mid x_i)\, v_i \qquad v_i = \begin{bmatrix} y_{2i} \\ y_{3i} \end{bmatrix}$$

$$\Sigma = \frac{1}{N_K} \cdot \sum_{i=0}^{m_0} p(K = \pi_K \mid x_i)\,(v_i - \mu_K)(v_i - \mu_K)^T$$

cluster 1:

$$P_1 = \frac{1}{1,54467}\left(0,19259 \cdot 1 + 0,63735 \cdot 0 + 0,55187 \cdot 0 + 0,16892 \cdot 1\right) = 0,23404$$

$$\mu_1 = \frac{1}{1,54467}\left(0,19259 \cdot \begin{bmatrix} 0,6 \\ 0,1 \end{bmatrix} + 0,63735 \cdot \begin{bmatrix} -0,4 \\ 0,8 \end{bmatrix} + 0,55187 \cdot \begin{bmatrix} 0,2 \\ 0,5 \end{bmatrix} + 0,16892 \cdot \begin{bmatrix} 0,4 \\ -0,1 \end{bmatrix}\right) = \begin{bmatrix} 0,02657 \\ 0,50713 \end{bmatrix}$$

$$\Sigma_1 = \frac{1}{1,54467} \cdot \Bigg(0,19259 \begin{bmatrix} 0,57349 \\ -0,40713 \end{bmatrix} \begin{bmatrix} 0,57349 & -0,40713 \end{bmatrix} + 0,63735 \begin{bmatrix} -0,42657 \\ 0,29287 \end{bmatrix} \begin{bmatrix} -0,42657 & 0,29287 \end{bmatrix}$$

$$+ 0,55187 \begin{bmatrix} 0,17349 \\ 0,00713 \end{bmatrix} \begin{bmatrix} 0,17349 & 0,00713 \end{bmatrix} + 0,16892 \begin{bmatrix} 0,37349 \\ -0,40713 \end{bmatrix} \begin{bmatrix} 0,37349 & -0,40713 \end{bmatrix}\Bigg)$$

$$= \begin{bmatrix} 0,14137 & -0,10541 \\ -0,10541 & 0,09605 \end{bmatrix}$$

cluster 2:

$$P_2 = 0,66732$$

$$\mu_2 = \begin{bmatrix} 0,30914 \\ 0,21042 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 0,10829 & -0,08865 \\ -0,08865 & 0,10412 \end{bmatrix}$$

2. Given the new observation, $x_{new} = (1, 0.3, 0.7)$, determine the cluster memberships (posteriors).

To determine the cluster memberships, We apply the E-STEP to the new observation, with the priors and parameters updated.

2 -

$$P(k = \pi_1 \mid x_{new}) = \frac{P(y1_{new} \mid k = \pi_1)\ P(y_{2\,new}\ y_{3\,new} \mid k = \pi_1)\ P(k = \pi_1)}{P(x_{new})}$$

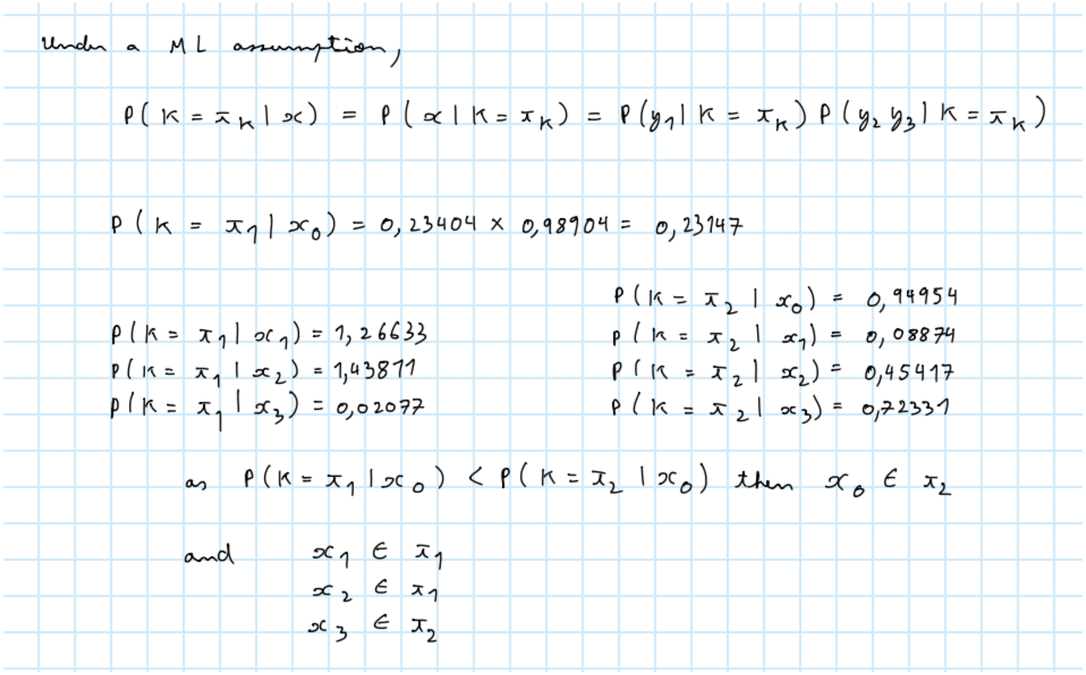
$$= \frac{0{,}23404 \times 0{,}02708 \times 0{,}38617}{P(x_{new})} = \frac{0{,}00245}{P(x_{new})} = 0{,}08029$$

$$P(k = \pi_2 \mid x_{new}) = \frac{0{,}02803}{P(x_{new})} = 0{,}91971$$

$$P(x_{new}) = 0{,}00245 + 0{,}02803 = 0{,}03048$$

3. Performing a hard assignment of observations to clusters under a ML assumption, identify the silhouette of both clusters under a Manhattan distance.

Under a ML assumption, we treat all priors as having equal probabilities, allowing us to disregard them. Similarly, we omit the probability of 'X.' In the classification of observations into the correct clusters, our primary concern is identifying the cluster with the highest probability rather than the precise probability value.



Under a ML assumption,

$$P(K=\pi_k \mid x) = P(x \mid K=\pi_k) = P(y_1 \mid K=\pi_k)\, P(y_2\, y_3 \mid K=\pi_k)$$

$$P(K=\pi_1 \mid x_0) = 0{,}23404 \times 0{,}98904 = 0{,}23147$$

$$P(K=\pi_2 \mid x_0) = 0{,}94954$$

$$P(K=\pi_1 \mid x_1) = 1{,}26633 \qquad P(K=\pi_2 \mid x_1) = 0{,}08874$$

$$P(K=\pi_1 \mid x_2) = 1{,}43871 \qquad P(K=\pi_2 \mid x_2) = 0{,}45417$$

$$P(K=\pi_1 \mid x_3) = 0{,}02077 \qquad P(K=\pi_2 \mid x_3) = 0{,}72331$$

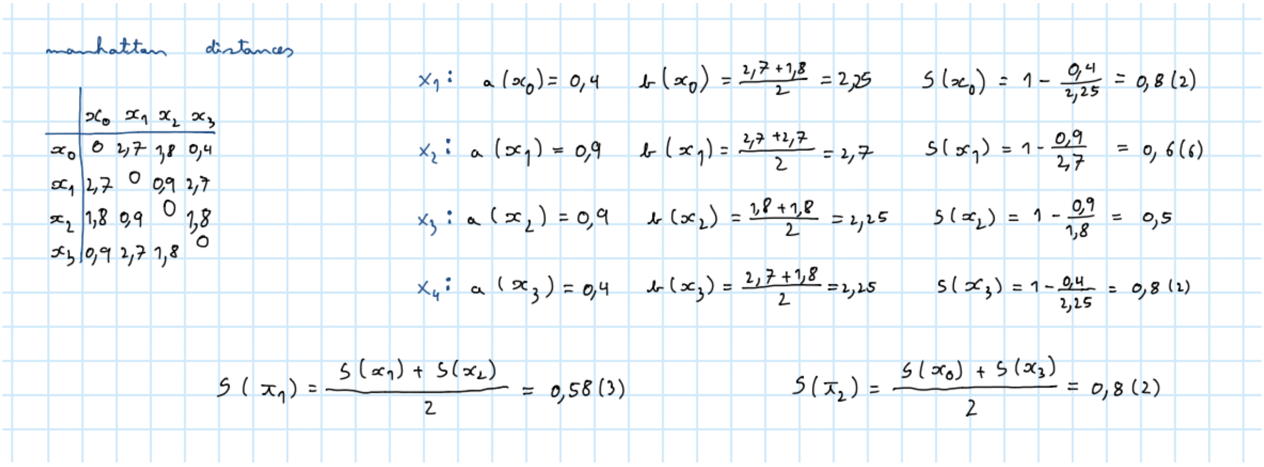
as $P(K=\pi_1 \mid x_0) < P(K=\pi_2 \mid x_0)$ then $x_0 \in \pi_2$

and $x_1 \in \pi_1$

$x_2 \in \pi_1$

$x_3 \in \pi_2$

After classifying the observations, in order to calculate the silhouette of the clusters, we start by calculating the Manhattan distances between them. Then the silhouette for each observation, and finally, we calculate the average of the observations within the clusters to obtain each cluster's silhouette.



manhattan distances

	x_0	x_1	x_2	x_3
x_0	0	2,7	1,8	0,4
x_1	2,7	0	0,9	2,7
x_2	1,8	0,9	0	1,8
x_3	0,9	2,7	1,8	0

X_1: $a(x_0) = 0{,}4 \quad b(x_0) = \frac{2{,}7+1{,}8}{2} = 2{,}25 \quad S(x_0) = 1 - \frac{0{,}4}{2{,}25} = 0{,}8(2)$

X_2: $a(x_1) = 0{,}9 \quad b(x_1) = \frac{2{,}7+2{,}7}{2} = 2{,}7 \quad S(x_1) = 1 - \frac{0{,}9}{2{,}7} = 0{,}6(6)$

X_3: $a(x_2) = 0{,}9 \quad b(x_2) = \frac{1{,}8+1{,}8}{2} = 2{,}25 \quad S(x_2) = 1 - \frac{0{,}9}{1{,}8} = 0{,}5$

X_4: $a(x_3) = 0{,}4 \quad b(x_3) = \frac{2{,}7+1{,}8}{2} = 2{,}25 \quad S(x_3) = 1 - \frac{0{,}4}{2{,}25} = 0{,}8(2)$

$$S(\pi_1) = \frac{S(x_1) + S(x_2)}{2} = 0{,}58(3) \qquad S(\pi_2) = \frac{S(x_0) + S(x_3)}{2} = 0{,}8(2)$$

4. Knowing the purity of the clustering solution is 0.75, identify the number of possible classes (ground truth)

Purity is a measure that tells us the quality of the clustering. The closer to 1, the better the clustering is.

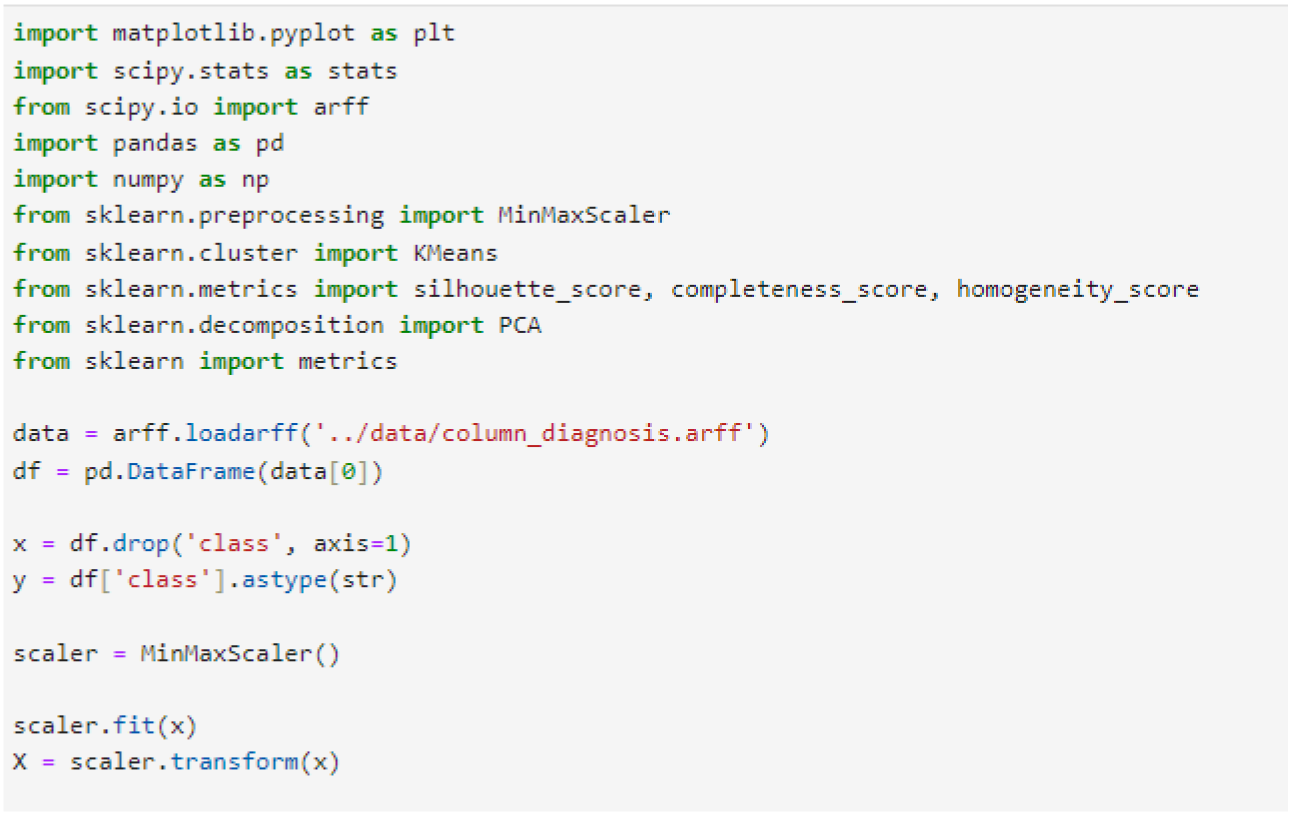
A purity value of 0.75 implies that 75% of the data points are correctly assigned to their respective clusters, while 25% are misclassified. In the context of having four observations, this means that only one of them is misclassified

In the case of this misclassified observation, there are two possible scenarios. It could either belong to the other existing cluster or belong to a new cluster, implying that it does not align with any of the pre-existing known classes.

Hence, we can deduce that there are, at the very least, two distinct classes, and potentially up to three classes when considering the possibility of the misclassified data point.

Part II: Programming

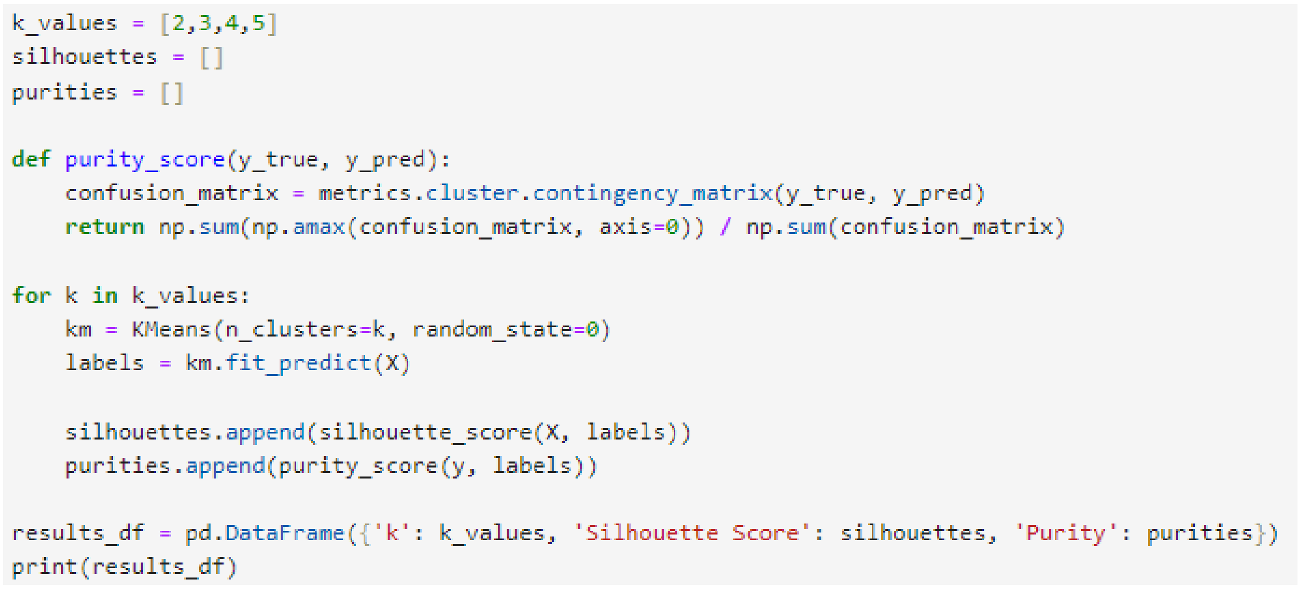
In the next image, we have provided an overview of all the imports utilized and our approach to importing the data set. Specifically, ’x’ stores all of the input variables values, and ’y’ stores the Output Class values. Capital ”X” stores the input variables after normalization.
To streamline the code, we’ve omitted this information from the beginning of each section to avoid redundancy.



```python
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.io import arff
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, completeness_score, homogeneity_score
from sklearn.decomposition import PCA
from sklearn import metrics

data = arff.loadarff('../data/column_diagnosis.arff')
df = pd.DataFrame(data[0])

x = df.drop('class', axis=1)
y = df['class'].astype(str)

scaler = MinMaxScaler()

scaler.fit(x)
X = scaler.transform(x)
```

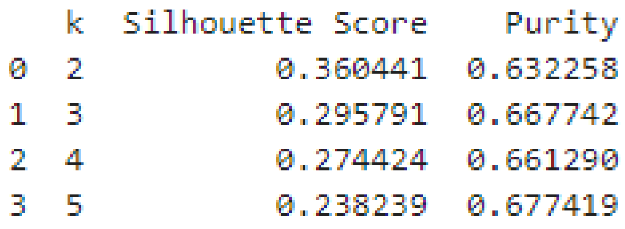
1. Apply k-means clustering fully unsupervisedly on the normalized data with $k \in \{2, 3, 4, 5\}$. Assess the silhouette and purity of the produced solutions.

Code:



```python
k_values = [2,3,4,5]
silhouettes = []
purities = []

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

for k in k_values:
    km = KMeans(n_clusters=k, random_state=0)
    labels = km.fit_predict(X)

    silhouettes.append(silhouette_score(X, labels))
    purities.append(purity_score(y, labels))

results_df = pd.DataFrame({'k': k_values, 'Silhouette Score': silhouettes, 'Purity': purities})
print(results_df)
```

Output:



```
   k  Silhouette Score    Purity
0  2          0.360441  0.632258
1  3          0.295791  0.667742
2  4          0.274424  0.661290
3  5          0.238239  0.677419
```

As we can see, with the increase of k, we get a decreasing value for the silhouette score (around 0.3). The decrease is explained by the fact that with more clusters the distances between them, b(x), gets smaller. The silhouette is inversely proportional to b(x).

Overall, from the average value of 0.3, we can assess observations in the same clusters are moderately similar between them.

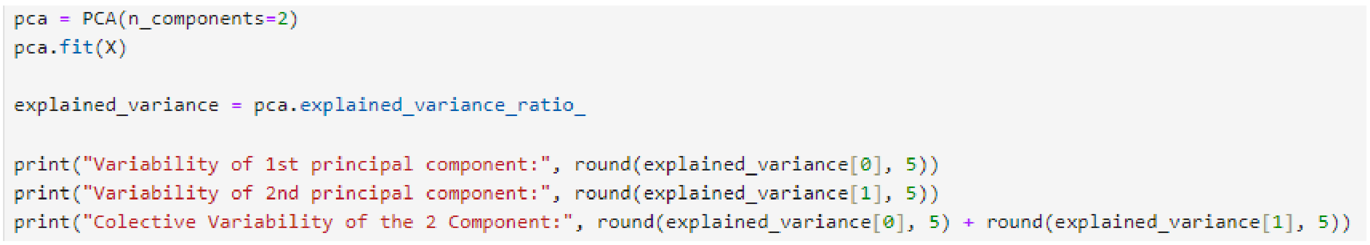
Regarding purity, the opposite happens. The more clusters there are, the better the purity (rounding 0.66). This is however prone to over-fitting, as, creating many, small/granular clusters will group the observations into very specific patterns that may not represent reality.

A purity of 0.66 indicates that most value are correctly assigned to the class.

2. Consider the application of PCA after the data normalization:

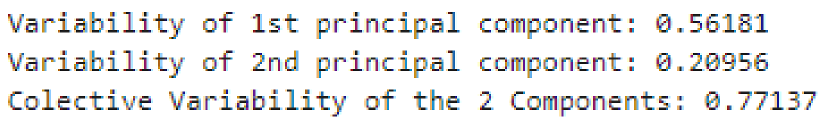
   (a) Identify the variability explained by the top two principal components.

   Code:



```
pca = PCA(n_components=2)
pca.fit(X)

explained_variance = pca.explained_variance_ratio_

print("Variability of 1st principal component:", round(explained_variance[0], 5))
print("Variability of 2nd principal component:", round(explained_variance[1], 5))
print("Colective Variability of the 2 Component:", round(explained_variance[0], 5) + round(explained_variance[1], 5))
```

   Output:

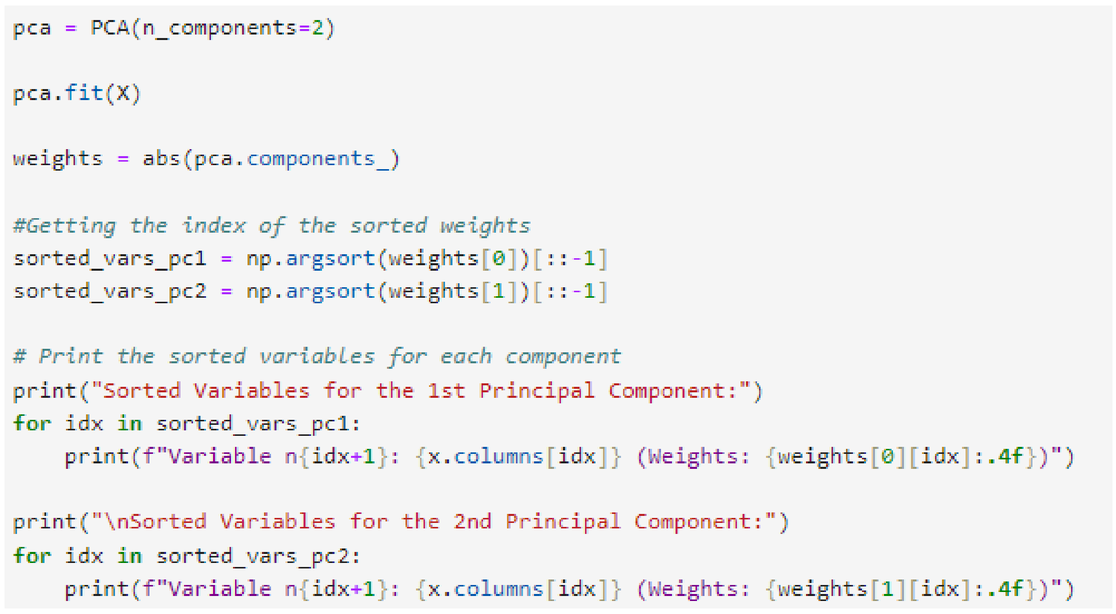


```
Variability of 1st principal component: 0.56181
Variability of 2nd principal component: 0.20956
Colective Variability of the 2 Components: 0.77137
```

   Overall, applying PCA with a dimensionality of 2 will retain 77.1% of the data.

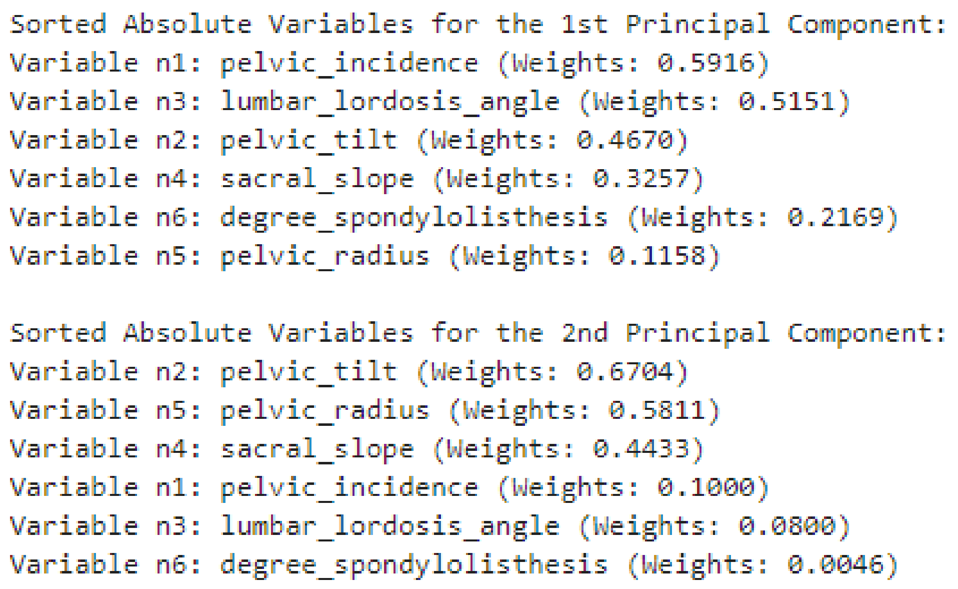
   (b) For each one of these two components, sort the input variables by relevance by inspecting the absolute weights of the linear projection.

   Code:



```
pca = PCA(n_components=2)

pca.fit(X)

weights = abs(pca.components_)

#Getting the index of the sorted weights
sorted_vars_pc1 = np.argsort(weights[0])[::-1]
sorted_vars_pc2 = np.argsort(weights[1])[::-1]

# Print the sorted variables for each component
print("Sorted Variables for the 1st Principal Component:")
for idx in sorted_vars_pc1:
    print(f"Variable n{idx+1}: {x.columns[idx]} (Weights: {weights[0][idx]:.4f})")

print("\nSorted Variables for the 2nd Principal Component:")
for idx in sorted_vars_pc2:
    print(f"Variable n{idx+1}: {x.columns[idx]} (Weights: {weights[1][idx]:.4f})")
```

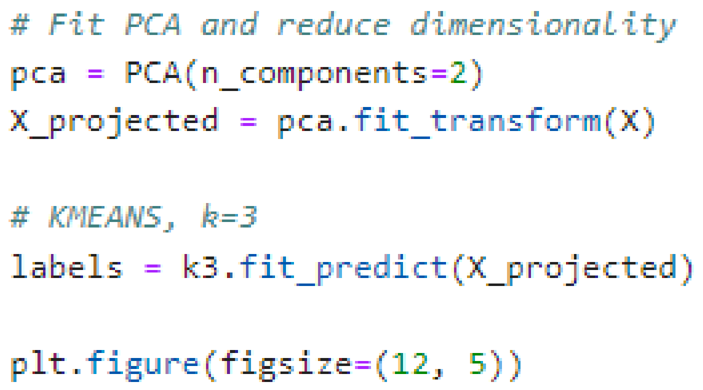
Output:



```
Sorted Absolute Variables for the 1st Principal Component:
Variable n1: pelvic_incidence (Weights: 0.5916)
Variable n3: lumbar_lordosis_angle (Weights: 0.5151)
Variable n2: pelvic_tilt (Weights: 0.4670)
Variable n4: sacral_slope (Weights: 0.3257)
Variable n6: degree_spondylolisthesis (Weights: 0.2169)
Variable n5: pelvic_radius (Weights: 0.1158)

Sorted Absolute Variables for the 2nd Principal Component:
Variable n2: pelvic_tilt (Weights: 0.6704)
Variable n5: pelvic_radius (Weights: 0.5811)
Variable n4: sacral_slope (Weights: 0.4433)
Variable n1: pelvic_incidence (Weights: 0.1000)
Variable n3: lumbar_lordosis_angle (Weights: 0.0800)
Variable n6: degree_spondylolisthesis (Weights: 0.0046)
```

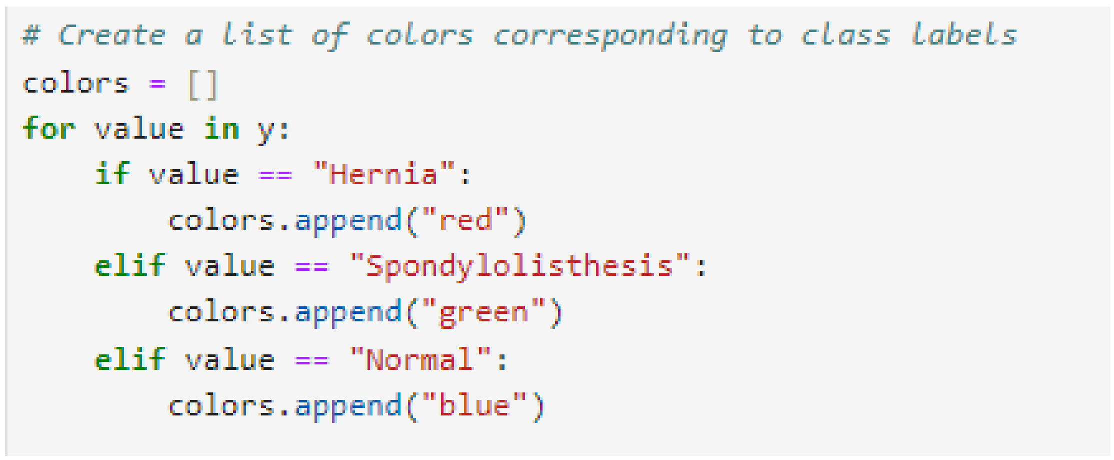
3. Project the normalized ground diagnoses, and the previously learned $k = 3$ clustering solution onto a 2-dimensional data space using PCA and then color observations using the reference and cluster annotations.

Projecting the data onto the 2 PCAs and calculating the clusters using k-means.



```
# Fit PCA and reduce dimensionality
pca = PCA(n_components=2)
X_projected = pca.fit_transform(X)

# KMEANS, k=3
labels = k3.fit_predict(X_projected)

plt.figure(figsize=(12, 5))
```

Because our output variable is categorical we must transform it to be numeric.

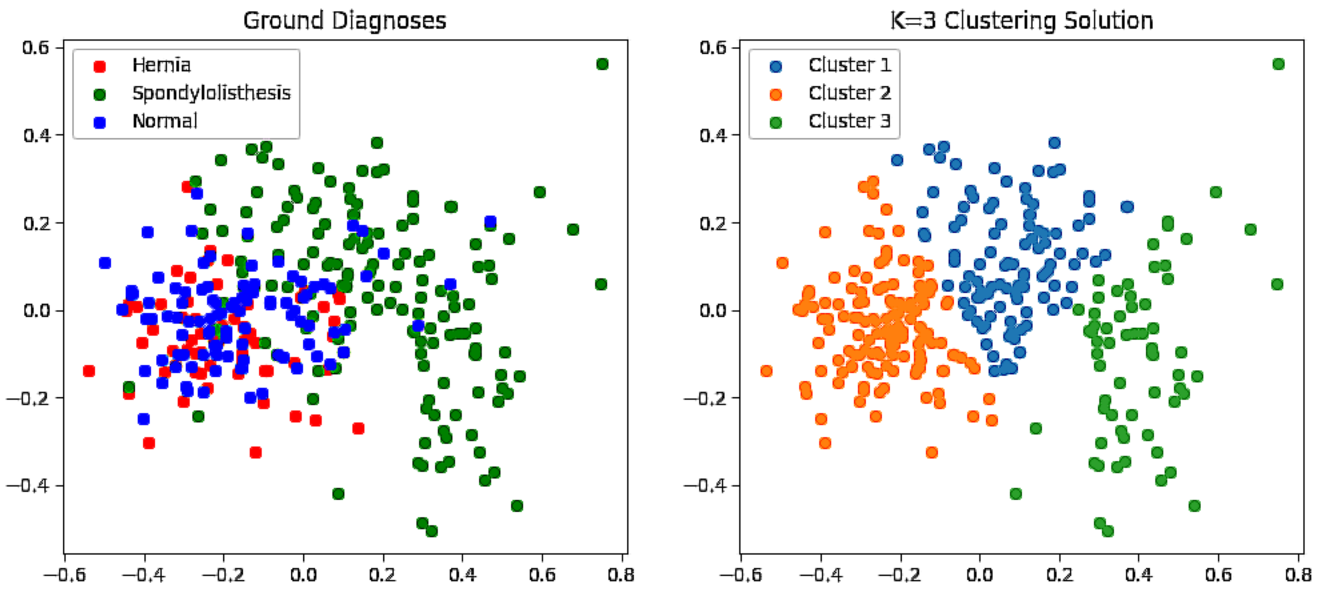


```
# Create a list of colors corresponding to class labels
colors = []
for value in y:
    if value == "Hernia":
        colors.append("red")
    elif value == "Spondylolisthesis":
        colors.append("green")
    elif value == "Normal":
        colors.append("blue")
```

Plotting the resulting clusters and the ground truth:

```
# Subplot for ground diagnoses
plt.subplot(1, 2, 1)
for label, col in zip(["Hernia", "Spondylolisthesis", "Normal"], ["red", "green", "blue"]):
    plt.scatter(X_projected[y == label, 0], X_projected[y == label, 1], c=col, label=label, cmap='viridis')
plt.title("Ground Diagnoses")
plt.legend()

# Subplot for the k=3 clustering solution
plt.subplot(1, 2, 2)
for cluster in range(3):
    plt.scatter(X_projected[labels == cluster, 0], X_projected[labels == cluster, 1], label=f'Cluster {cluster + 1}', cmap='viridis')
plt.title("K=3 Clustering Solution")
plt.legend()

plt.show()
```

Output:



4. 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.

Two ways on how clustering can be used to characterize the population of ill and healthy individuals are Differentiation of Health Status and Anomaly Detection.

Differentiation of Health Status:

Clustering algorithms can help identify subgroups or clusters within the population of ill individuals. By analyzing the graph in the exercise 3, we can identify groups of individuals with similar health profiles. For instance, there is a strong indication that individuals belonging to cluster 3 exhibit a very high likelihood of being diagnosed with Spondylolisthesis.

These indications are reinforced by the results from exercise 1, where k-means scored a purity of (0.667), which suggests that the algorithm effectively groups data points with others from the same class, demonstrating a reasonable level of separation.

Anomaly Detection:

Clustering is also a great way to detect anomalies or outliers in a certain population. In the context of healthcare, anomalies may represent individuals who have unusual or unexpected health conditions. By comparing the clusters assigned to the ground truth diagnoses (as in question 3), we can identify individuals who are assigned to clusters that do not align with their actual diagnoses. These individuals may require further investigation as they could have undiagnosed or rare health conditions, making clustering a valuable tool for early disease detection.