Aprendizado de Máquina
Lista 03

Ricardo Marra - 19/0137576

I. Questão 01

Consider the three-dimensional dataset in dados_lista3.csv. Apply PCA to study the data, writing your own code.

A. What is the sample mean of the data? Then, create a new dataset with null sample mean.

Para este problema, denominou-se o conjunto de dados **X**. Formado pelos três atributos **X1**, **X2** e **X3**. A média de cada atributo é mostrada na Tabela I.

TABLE I
MÉDIA DOS DADOS

X1	X2	X3
2.523896	−2.812758	0.303879

Para se construir um novo conjunto de dados com média nula, é subtraído de cada atributo o valor de sua média. Além disso, neste caso, cada atributo foi divido por seu desvio padrão. Este processo é dito como normalização dos dados, fazendo com que todo o conjunto de dados tenha média nula e desvio padrão igual a um. A formulação matemática é dada por (1).

$$X_{norm} = \frac{X - avg(X)}{std(X)} \tag{1}$$

B. Calculate the sample covariance matrix of the null-mean dataset. Calculate its eigenvalues and eigenvectors.

A partir dos dados normalizados, é possível construir a matriz de correlação Σ, visto em (2).

$$\Sigma = \begin{bmatrix} 1 & -0.533858 & -0.606122 \\ -0.533858 & 1 & 0.940140 \\ -0.606122 & 0.940140 & 1 \end{bmatrix} \tag{2}$$

Com a matriz de correlação é possível, então, calcular seus autovalores (3) e autovetores (4).

$$\alpha = \begin{bmatrix} 2.40348266 & 0.54101145 & 0.05550589 \end{bmatrix} \tag{3}$$

$$w = \begin{bmatrix} 0.49831409 & -0.86283557 & 0.08484009 \\ -0.60532847 & -0.41630393 & -0.6784309 \\ -0.62069357 & -0.28671556 & 0.72974905 \end{bmatrix} \tag{4}$$

C. Based on the results of (a) and (b), analyze if it is possible to reduce the dataset to (i) one or (ii) two dimensions. Use numerical measures to justify your analysis.

A partir dos autovalores da matriz de correlação é possível calcular a Proporção de Variância (PdV), de acordo com a equação (5). A proporção de variância diz o quanto da variância do conjunto de dados completo é explicada pelo conjunto de dados reduzido.

$$PdV = \frac{\lambda_1 + \lambda_2 + ... + \lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_k + ... + \lambda_d} \tag{5}$$

O conjunto de dados tem 3 atributos, portanto $d = 3$. A análise então é feita para $k = 1$ e $k = 2$. A PdV de cada dimensão pode ser vista na Tabela II. Neste caso, um autovetor representa cerca de 80% da variância do conjunto de dados completo, já dois autovetores representam cerca de 98% da variância.

TABLE II
PROPORÇÃO DE VARIÂNCIA

Autovetores	PdV
1	0.80116089
2	0.98149804

Em casos onde se tem vários atributos, utilizar 80% da variância pode ser uma boa aproximação. Porém, tipicamente se utiliza uma quantidade de autovetores em que se explicam cerca de 90% do conjunto de dados.

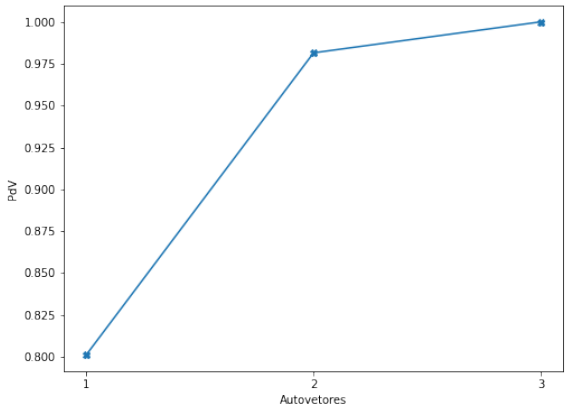

Fig. 1. Gráfico da Proporção de Variância pela quantidade de autovetores.

Uma forma de escolher o número de autovetores, é utilizar o gráfico da PdV pelo número de autovetores e escolher o valor no ”cotovelo”, antes do gráfico crescer linearmente. Como pode se observar na Fig. 1, o ”cotovelo” acontece utilizando dois autovetores.

É possível utilizar tanto um, quanto dois autovetores para representar o conjunto de dados. Porém, a representação com apenas um autovetor pode não ser próxima da realidade. Portanto, utilizar dois autovetores, neste caso, se torna a melhor opção.

D. Plot, in the same 3D graph: (i) the original data, (ii) the reconstructed data from the 1D projection and (iii) the reconstructed data from the 2D projection. Comment the results.

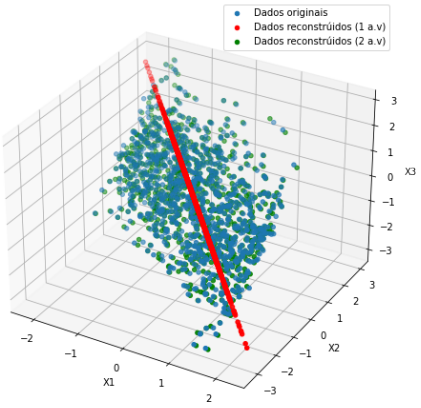

Fig. 2. Gráfico de dispersão dos dados originais (azul), reconstruídos da projeção 1D (vermelho) e reconstruídos da projeção 2D (verde).

Como dito em I-C, a reconstrução com apenas um autovetor não é uma boa representação do conjunto de dados. Além de se observar isso no gráfico de sua reconstrução, comparando os valores de desvio padrão dos dados originais e reconstruídos na Tabela III, percebe-se que há uma perda considerável de variância.

TABLE III
DESVIOS PADRÕES DAS RECONSTRUÇÕES

Projeção	**X1**	**X2**	**X3**
Original	1	1	1
1D	0.77215842	0.93798164	0.96179051
2D	0.99930019	0.98664985	0.98461708

Por outro lado, a reconstrução da projeção 2D é quase idêntica aos dados originais, como era de se esperar, visto que a perda é quase nula.

E. Based on the previous results, is PCA a useful tool for this dataset?

Devido ao fato de que, utilizando apenas $\frac{2}{3}$ dos atributos, se reconstrói o conjunto de dados explicando 98% da variância do original, a aplicação de PCA se torna sim uma ferramenta útil para este conjunto de dados em específico.

II. QUESTÃO 02

A. Choose a 2D or 3D dataset to perform k-means clustering. Third-party libraries / toolboxes are allowed. Explain the steps of your solution and provide a justified decision on the number of clusters (graphically or numerically).

O conjunto de dados utilizado para aplicar k-means foi o dataset **iris**. O conjunto de dados tem quatro atributos: largura e comprimento da sépala, largura e comprimento da pétala. Para melhor visualização dos dados, escolheu-se três dos quatro atributos. A escolha foi feita a partir da correlação do atributo com o valor alvo (tipo da íris). Portanto, excluiu-se a largura da sépala, visto que se obteve a menor correlação. Os atributos podem ser observados na Fig. 3.

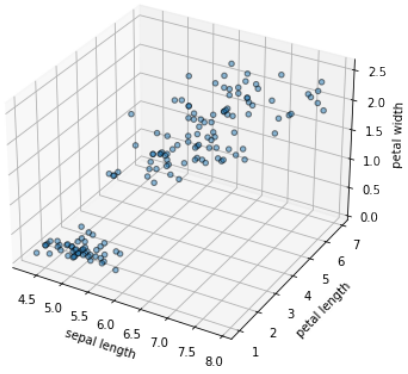

Fig. 3. Gráfico de dispersão dos atributos.

Para se determinar o número de clusters utilizados no algoritmo de k-means, se utiliza de uma abordagem chamada método do cotovelo.

Para se explicar este método, primeiro deve-se comentar sobre a inércia na clusterização. A inércia é a soma da diferença ao quadrado entre os pontos e o centroide do cluster, dada pela equação (6). Quanto menor a inércia, mais coerente a classificação do cluster.

$$I = \sum_{i=1}^{N} (x_i - C_k)^2 \tag{6}$$

O método do cotovelo consiste na confecção do gráfico da inércia pelo número de clusters. A escolha do melhor número de clusters é dado pelo último valor de k (número de clusters) onde a inércia ainda não começou a decair linearmente, ou

seja, no ”cotovelo” do gráfico. O gráfico para este caso pode ser visto na Fig. 4.

Escolheu-se valores de k entre 1 e 10. Para cada k, aplicou-se k-means no conjunto de dados e calculou-se sua inércia.

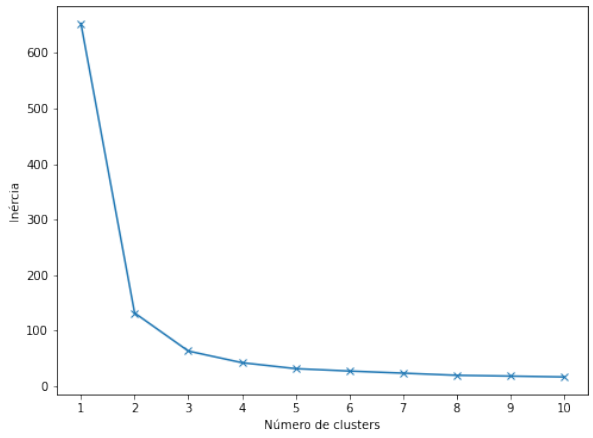

Fig. 4. Gráfico da inércia pelo número de clusters.

Como pode-se perceber na Fig. 4, o ponto antes do decaimento da inércia se encontra em $k = 3$, o ”cotovelo”. Portanto, treina-se o modelo utilizando três clusters. O resultado das amostras classificadas pode ser visto na Fig. 5, onde se plotou todos os atributos com suas classificações feitas pelo modelo de k-means.

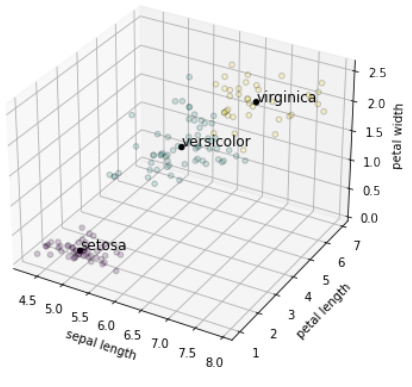

Fig. 5. Amostras classificadas.

Na imagem é possível ver os centroides de cada cluster (os pontos em preto), bem como sua classificação, dada pela cor de cada amostra classificada e sua respectiva classificação.

Ao se observar a Fig. 5, percebe-se que a escolha de $k = 3$ faz sentido, visto que dois clusters concentraria muito uma parte dos dados e, pela distribuição, não seria necessário um quarto cluster.

III. Questão 03

Consider the following table, which shows the geodesic distance (km) between some Brazilian capital cities:

	Brasília	Manaus	Natal	São Paulo	Porto Alegre	Rio de Janeiro	Rio Branco
Brasília	0	1950	1775	872	1618	931	2256
Manaus	1950	0	2760	2675	3132	2838	1145
Natal	1775	2760	0	2322	3168	2086	3617
São Paulo	872	2675	2322	0	858	360	2694
Porto Alegre	1618	3132	3168	858	0	1116	2814
Rio de Janeiro	931	2838	2086	360	1116	0	2991
Rio Branco	2256	1145	3617	2694	2814	2991	0

Fig. 6. Distâncias geodésicas.

A. Apply Multidimensional Scaling to create and plot a 2D map of the cities. Compare your results with the real map of Brazil. Third-party libraries / toolboxes are allowed. Explain the steps of your solution.

O escalonamento multidimensional é uma técnica utilizada para visualizar a similaridade entre atributos a partir de uma matriz de dissimilaridade. A partir da matriz, o algoritmo projeta a distância entre os atributos em um plano cartesiano 2D ou 3D.

Neste caso, o conjunto de dados consiste em distâncias entre cidades, que são facilmente traduzidas em pontos em um mapa bidimensional. Portanto, aplica-se o algoritmo em duas componentes (mapa 2D).

Como a matriz de dissimilaridade já é o próprio conjunto de dados, não é necessário realizar nenhuma transformação. O único ponto a se detalhar é que foi necessário fazer os atributos índices da matriz, para que cada ponto da matriz fosse uma distância entre atributos.

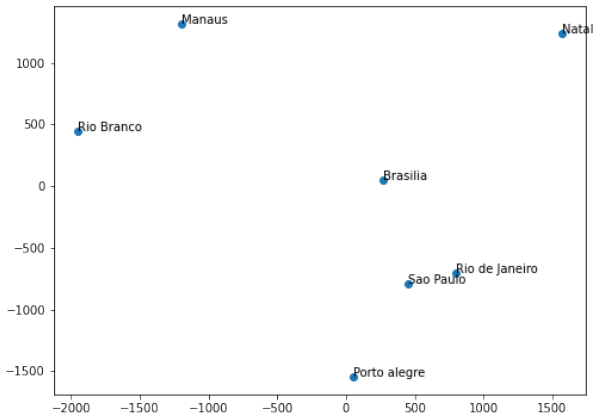

Fig. 7. Mapa com as cidades.

O resultado em plano cartesiano pode ser visto na Fig. 7. Para título de comparação, temos na Fig. 8 um mapa do Brasil com as capitais de cada estado marcadas.

Como não se tem informação de nenhuma capital acima de Manaus, o mapa termina ali. Porém, é possível observar que as outras capitais estão espaçadas de forma muito semelhante ao mapa brasileiro.

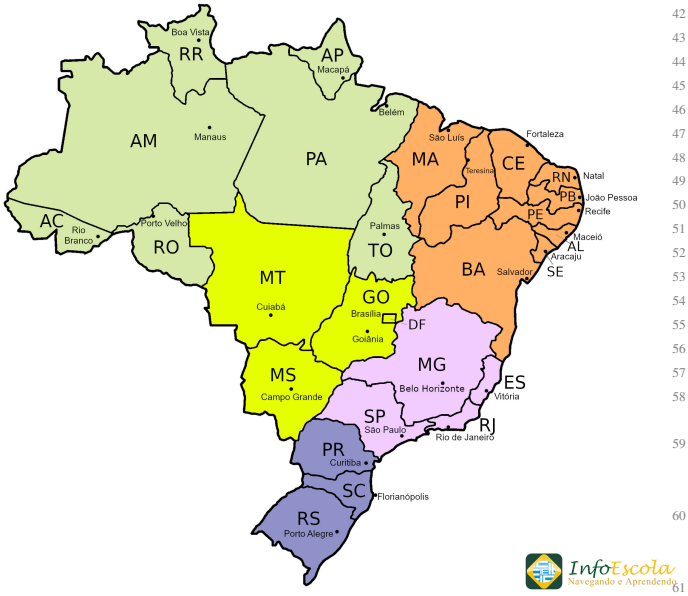

Fig. 8. Mapa do Brasil.

IV. Apêndice

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def normalize(df: pd.DataFrame) -> pd.DataFrame:

    df_norm = (df - df.mean())/df.std()

    return df_norm

def pca(X: pd.DataFrame, n_dim: int) -> tuple:

    a, w = np.linalg.eig(X.corr())

    idx_sorted = np.flip(np.argsort(a))

    W = np.zeros([X.shape[1], n_dim])

    for i in range(n_dim):
        W[:, i] = w[:, idx_sorted[i]]

    z = np.dot(W.T, X.T)

    return W, z.T

data = pd.read_csv('data/dados_lista3.csv', header = None, names = ['X1', 'X2', 'X3'])
print(data.mean())

data_norm = normalize(data)
print(data_norm.corr())

a, w = np.linalg.eig(data_norm.corr())
print(a)
print(w)

pov = np.cumsum(a)/np.sum(a)
fig = plt.figure()
fig.set_size_inches(8,6)
fig.set_dpi(80)

plt.plot([1, 2, 3], pov, marker = 'X')
plt.xlabel('Autovetores')
plt.ylabel('PdV')
plt.xticks([1, 2, 3])
plt.savefig('images/plotq1c')
plt.show()

W1, z1 = pca(data_norm, 1)
W2, z2 = pca(data_norm, 2)

X1_reconstructed = np.dot(W1, z1.T).T
X2_reconstructed = np.dot(W2, z2.T).T

fig = plt.figure()
fig.set_size_inches(8,6)
fig.set_dpi(300)
ax = fig.add_subplot(projection = '3d')
ax.scatter(data_norm['X1'], data_norm['X2'], data_norm['X3'], label = 'Dados originais')
ax.scatter(X1_reconstructed[:, 0], X1_reconstructed[:, 1], X1_reconstructed[:, 2], color = 'red', label = 'Dados reconstridos (1 a.v)')
ax.scatter(X2_reconstructed[:, 0], X2_reconstructed[:, 1], X2_reconstructed[:, 2], color = 'green', label = 'Dados reconstridos (2 a.v)')
ax.set_xlabel('X1')
ax.set_ylabel('X2')
ax.set_zlabel('X3')

plt.legend(loc = 'upper right')
plt.savefig('images/plotq1d')
plt.show()
```

Listing 1. Códigos referente a Questão 01

```python
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans

data = load_iris()

X = pd.DataFrame(data['data'], columns = data['feature_names'])
X.drop(columns = 'sepal width (cm)', inplace = True)
y = pd.DataFrame(data['target'], columns = ['target'])

mapping = {i : data['target_names'][i] for i in range(y['target'].max() + 1)}

fig = plt.figure()
fig.set_size_inches(8,6)
fig.set_dpi(80)
ax = fig.add_subplot(projection = '3d')
ax.scatter(X['sepal length (cm)'], X['petal length (cm)'], X['petal width (cm)'], alpha = 0.5, edgecolor= 'k')

ax.set_xlabel('sepal length')
ax.set_ylabel('petal length')
ax.set_zlabel('petal width')

plt.savefig('images/plotq2a')
plt.show()

clusters = range(1, 11)
inertia = {}

for k in clusters:

    model = KMeans(n_clusters = k, random_state = 0)
    model.fit(X)

    inertia[k] = model.inertia_

fig = plt.figure()
```

```
fig.set_size_inches(8,6)
fig.set_dpi(80)
plt.plot(inertia.keys(), inertia.values(), 'x-')
plt.xlabel('N mero de clusters')
plt.ylabel('In rcia')
plt.xticks(range(1,11))
plt.savefig('images/plotq2b')
plt.show()

best_model = KMeans(n_clusters = 3, random_state =
    0)
best_model.fit(X)

centroids = best_model.cluster_centers_
labels = best_model.labels_

fig = plt.figure()
fig.set_size_inches(8,6)
fig.set_dpi(80)
ax = fig.add_subplot(projection = '3d')
ax.scatter(X['sepal length (cm)'], X['petal length (
    cm)'], X['petal width (cm)'], c = labels.astype(
    float), alpha = 0.2, edgecolor= 'k')
ax.scatter(centroids[:, 0], centroids[:, 1],
    centroids[:, 2], c = 'k', alpha = 1, edgecolor =
     'k')

for i in range(len(centroids)):
    ax.text(centroids[i, 0], centroids[i, 1],
    centroids[i, 2], mapping[i], size = 12, zorder =
     1)

ax.set_xlabel('sepal length')
ax.set_ylabel('petal length')
ax.set_zlabel('petal width')
plt.savefig('images/plotq2c')
plt.show()
```

Listing 2. Códigos referente a Questão 02

```
from sklearn.manifold import MDS
cities = pd.read_csv('data/tabela_questao3.csv', sep
    = ';', index_col = 0)
mds_model = MDS(n_components = 2, random_state = 0,
    dissimilarity = 'precomputed')
mds_coordinates = mds_model.fit_transform(cities)

fig = plt.figure()
fig.set_size_inches(8,6)
fig.set_dpi(80)

labels = cities.columns
plt.scatter(mds_coordinates[:, 0], mds_coordinates
    [:, 1])

for label, x, y, in zip(labels, mds_coordinates[:,
    0], mds_coordinates[:, 1]):
    plt.annotate(label, (x, y))

plt.savefig('images/plotq3a')
plt.show()
```

Listing 3. Códigos referente a Questão 03