

# Álgebra Linear - PCA

Fernanda Rafaela

Novembro, 2024

## 1 Dataset

O dataset inclui fatores de risco que auxiliam na previsão e prevenção de doenças cardiovasculares, possibilitando a análise de elementos que influenciam a mortalidade por insuficiência cardíaca.

Link: <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data>

## 2 Desenvolvimento

Para realizar a normalização dos dados e aplicar o algoritmo de PCA, serão usadas as seguintes bibliotecas no Python:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 1: Bibliotecas utilizadas para análise

Primeiro, importamos o arquivo CSV e normalizamos o dataset, calculando a média de cada coluna e subtraindo do total:

```
6 df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
7 df_scaled = (df - df.mean()) / df.std()
8
```

	age	anaemia	creatinine_phosphokinase	...	smoking	time	DEATH_EVENT
0	1.190949	-0.869647	0.000165	...	-0.686531	-1.626775	1.451727
1	-0.490457	-0.869647	7.502063	...	-0.686531	-1.601007	1.451727
2	0.350246	-0.869647	-0.449186	...	1.451727	-1.588122	1.451727
3	-0.910808	1.146046	-0.485257	...	-0.686531	-1.588122	1.451727
4	0.350246	1.146046	-0.434757	...	-0.686531	-1.575238	1.451727

[5 rows x 13 columns]

Figure 2: Importação do CSV

Em seguida, calculamos a matriz de covariância, transpondo-a:



Figure 3: Matriz de Covariância

Agora, calculamos os autovetores e autovalores e ordenamos do maior para o menor:

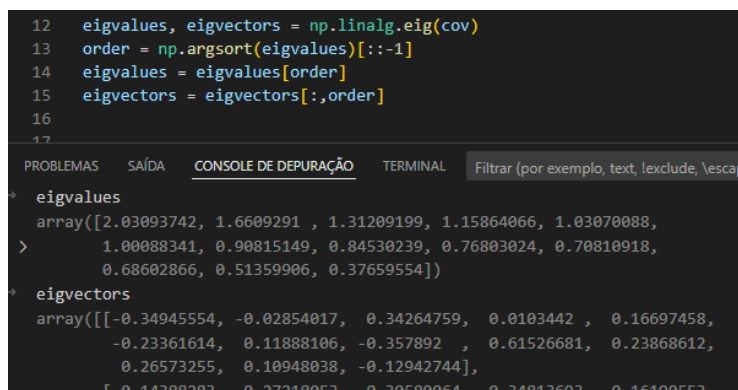


Figure 4: Cálculo dos Autovalores e Autovetores

Maiores autovalores e autovetores:

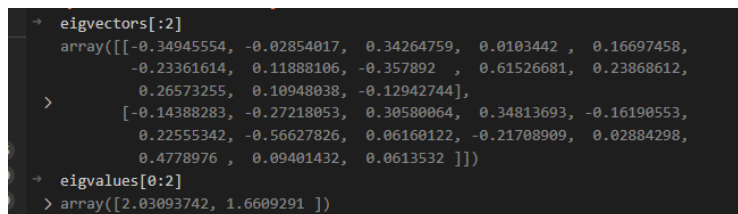


Figure 5: Exibição dos maiores autovetores e autovalores

Usando os 2 maiores autovalores calculamos o pca e fazemos um grafico bidimensional:

```
21 k = 2
22 pca = np.matmul(df_scaled, eigvectors[:, :k])
23
24 plt.figure(figsize=(8, 6))
25 sns.scatterplot(x=pca.iloc[:, 0], y=pca.iloc[:, 1])
26 plt.title('PCA: PC1 vs PC2')
27 plt.xlabel('Componente Principal 1')
28 plt.ylabel('Componente Principal 2')
29 plt.show()
```

Figure 6: Criação mapa 2d

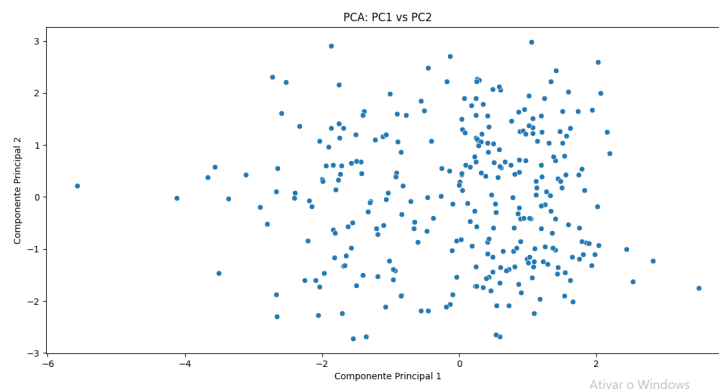


Figure 7: Mapa 2d

Usando os maiores 3 autovetores calculamos o pca e fazemos o gráfico tridimensional:

```
k = 3
pca = np.matmul(df_scaled, eigvectors[:, :k])

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(pca.iloc[:, 0], pca.iloc[:, 1], pca.iloc[:, 2])
ax.set_xlabel('Componente Principal 1')
ax.set_ylabel('Componente Principal 2')
ax.set_zlabel('Componente Principal 3')
plt.title('PCA: PC1 vs PC2 vs PC3')
plt.show()
```

Figure 8: Criação mapa 3d

PCA: PC1 vs PC2 vs PC3

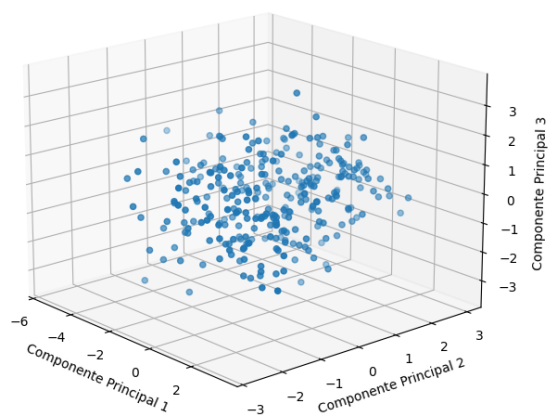


Figure 9: Mapa 3d