

November 9, 2023

```
[1]: import pandas as pd
import numpy as np
import time
```

```
[2]: file = 'ionosphere.data'
data = pd.read_csv(file, header=None)
# removendo a ultima coluna dos dados
# The 35th attribute is either "good" or "bad" according to the definition
# → summarized above. This is a binary classification task.
data = data.iloc[:, :-1]
print(data.shape)
```

(351, 34)

## 1 Matriz de covariância implementada

```
[6]: start = time.time()
mean_vector = np.mean(data, axis=0)
#print(mean_vector.shape)
centered_data = data - mean_vector
#print(centered_data.shape)
N = data.shape[0]
#print(N)
Cmy = (centered_data.T @ centered_data) / N
#print(Cmy.shape)
stop = time.time()

tempo_gasto = stop - start
print(f"Tempo gasto: {tempo_gasto} segundos")
```

Tempo gasto: 0.0029296875 segundos

## 2 Matriz de Covariância nativa

```
[7]: start2 = time.time()
Cref = np.cov(data, rowvar=False)
#print(Cref.shape)
```

```
stop2 = time.time()
tempo_gasto2 = stop2 - start2
print(f"Tempo gasto: {tempo_gasto2} segundos")
```

Tempo gasto: 0.0009810924530029297 segundos

O método nativo da linguagem se mostrou aproximadamente duas vezes mais rápido que o método implementado.