

November 9, 2023

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

```
[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]
#data = np.random.rand(30, 5)
print(data.shape)
```

(351, 34)

```
[3]: # Calcular a média de cada variável
mean_vector = np.mean(data, axis=0)
print(mean_vector.shape)
```

(34,)

```
[4]: # Centralizando os dados
centered_data = data - mean_vector
#print(centered_data)
```

```
[5]: N = data.shape[0]
#print(N)
c = (centered_data.T @ centered_data) / N
#print(c)
```

```
[6]: rank = np.linalg.matrix_rank(c)
print("O posto da matriz é:", rank)
```

O posto da matriz é: 33

Considerando que o posto da matriz não é completo (33), a matriz não é invertível, conforme tentativa abaixo.

```
[7]: # Calcule a inversa da matriz de covariância regularizada
inverse_matrix = np.linalg.inv(c)
```

```

-----
LinAlgError                                Traceback (most recent call last)
<ipython-input-7-5df52234e4fa> in <module>
      1 # Calcule a inversa da matriz de covariância regularizada
----> 2 inverse_matrix = np.linalg.inv(c)

~\AppData\Local\Programs\Python\Python38\lib\site-packages\numpy\core\overrides.py
↳ in inv(*args, **kwargs)

~\AppData\Local\Programs\Python\Python38\lib\site-packages\numpy\linalg\linalg.py
↳ in inv(a)
      543     signature = 'D->D' if isComplexType(t) else 'd->d'
      544     extobj = get_linalg_error_extobj(_raise_linalgerror_singular)
--> 545     ainv = _umath_linalg.inv(a, signature=signature, extobj=extobj)
      546     return wrap(ainv.astype(result_t, copy=False))
      547

~\AppData\Local\Programs\Python\Python38\lib\site-packages\numpy\linalg\linalg.py
↳ in _raise_linalgerror_singular(err, flag)
      86
      87 def _raise_linalgerror_singular(err, flag):
--> 88     raise LinAlgError("Singular matrix")
      89
      90 def _raise_linalgerror_nonposdef(err, flag):

LinAlgError: Singular matrix

```

0.1 Utilizaremos a técnica Variante 1 discutida em sala de aula e disponibilizada no pdf das aulas

```

[8]: # Ruído
lambda_value = 0.1
# cria uma matriz identidade de dimensões centered_data.shape[1]
identity_matrix = np.eye(centered_data.shape[1])
# criando matriz de regularização
regularization_matrix = lambda_value * identity_matrix

```

```

[9]: # Calculando a matriz de covariância regularizada adicionando ruído à matriz
↳ diagonal, tornando-a diagonalmente dominante.
regularized_cov_matrix = c + regularization_matrix

```

```

[10]: rank = np.linalg.matrix_rank(regularized_cov_matrix)
print("0 posto da matriz é:", rank)

```

0 posto da matriz é: 34

```
[11]: # Calculando a inversa da matriz de covariância regularizada
inverse_matrix = np.linalg.inv(regularized_cov_matrix)
inverse_matrix.shape
```

```
[11]: (34, 34)
```

```
[13]: # Calculando a SVD da matriz de covariância regularizada
U, S, VT = np.linalg.svd(regularized_cov_matrix)

# Inverte a matriz diagonal de autovalores (S)
inverse_S = 1.0 / S

# Calcula a matriz de covariância invertida
inverse_regularized_cov_matrix = np.dot(U, np.dot(np.diag(inverse_S), VT))
inverse_regularized_cov_matrix
```

```
[13]: array([[ 5.96105835e+00,  5.43548647e-16, -5.92826603e-01, ...,
           3.03878435e-01, -5.42712642e-02, -1.72631011e-01],
          [ 5.24289188e-15,  1.00000000e+01, -4.39183339e-15, ...,
           -7.53459793e-16, -5.50741269e-15,  7.82858706e-16],
          [-5.92826603e-01, -8.33362997e-17,  4.26007809e+00, ...,
           7.65138936e-02,  2.23571344e-02,  3.44410952e-02],
          ...,
          [ 3.03878435e-01,  3.17589996e-15,  7.65138936e-02, ...,
           4.14401761e+00, -9.09773041e-02, -9.31521797e-01],
          [-5.42712642e-02, -4.75588339e-15,  2.23571344e-02, ...,
           -9.09773041e-02,  5.09073178e+00,  3.67585216e-01],
          [-1.72631011e-01, -3.26955175e-15,  3.44410952e-02, ...,
           -9.31521797e-01,  3.67585216e-01,  4.44298608e+00]])
```