

Universidade Federal do Rio de Janeiro  
Escola Politécnica  
Departamento de Engenharia Eletrônica e de Computação  
Engenharia Eletrônica e de Computação  
Álgebra Linear 2

# **TRABALHO DE ÁLGEBRA LINEAR 2**

Aluna: Karen dos Anjos Arcoverde

Professor: Marcello Luiz Rodrigues de Campos

Rio de Janeiro  
2021

# Sumário

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>0</b> | <b>Introdução</b>              | <b>3</b>  |
| 0.1      | Conteúdo . . . . .             | 3         |
| 0.2      | Software e linguagem . . . . . | 3         |
| 0.3      | Bibliotecas . . . . .          | 3         |
| 0.4      | Base de dados . . . . .        | 3         |
| <b>1</b> | <b>Observações</b>             | <b>3</b>  |
| <b>2</b> | <b>Questão 1</b>               | <b>3</b>  |
| 2.1      | Resultados . . . . .           | 3         |
| 2.1.1    | Iris-Setosa . . . . .          | 3         |
| 2.1.2    | Iris-Versicolor . . . . .      | 4         |
| 2.1.3    | Iris-Virginica . . . . .       | 4         |
| <b>3</b> | <b>Questão 2</b>               | <b>4</b>  |
| 3.1      | Resultados . . . . .           | 4         |
| 3.1.1    | Iris-Setosa . . . . .          | 4         |
| 3.1.2    | Iris-Versicolor . . . . .      | 5         |
| 3.1.3    | Iris-Virginica . . . . .       | 6         |
| <b>4</b> | <b>Questão 3</b>               | <b>6</b>  |
| 4.1      | Resultados . . . . .           | 6         |
| 4.1.1    | Iris-Setosa . . . . .          | 6         |
| 4.1.2    | Iris-Versicolor . . . . .      | 7         |
| 4.1.3    | Iris-Virginica . . . . .       | 8         |
| <b>5</b> | <b>Questão 4</b>               | <b>9</b>  |
| 5.1      | Resultados . . . . .           | 9         |
| <b>6</b> | <b>Código</b>                  | <b>9</b>  |
| <b>7</b> | <b>Bibliografia</b>            | <b>19</b> |

# 0 Introdução

## 0.1 Conteúdo

O relatório contém os resultados encontrados para cada questão passada pelo professor e o código final em linguagem Python.

## 0.2 Software e linguagem

O software usado para programação foi o Spyder e a linguagem foi o Python 3.8.

## 0.3 Bibliotecas

As bibliotecas utilizadas para construir o código em Python foram:

- Numpy
- Scipy

## 0.4 Base de dados

O conjunto de dados "Iris" selecionado para o trabalho foi:

| ESPÉCIES        | DE  | PARA |
|-----------------|-----|------|
| Iris-setosa     | 25  | 39   |
| Iris-versicolor | 75  | 89   |
| Iris-virginica  | 125 | 139  |

Tabela 1: Base de dados "Iris" selecionada

# 1 Observações

A variável  $y$  da coluna PetalWidthCm foi escrita em função das outras variáveis  $x_1, x_2, x_3$  das colunas SepalLengthCm, SepalWidthCm, PetalLengthCm, respectivamente. De forma que  $y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3$  sem o termo independente. Com o termo independente:  $y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + k$ .

Além disso, a equação normal é:  $x^T \cdot x \cdot w = x^T \cdot y$ , onde  $w$  é o vetor de coeficientes, o vetor  $y$  é a coluna PetalWidthCm e a matriz  $x$  é as colunas SepalLengthCm, SepalWidthCm, PetalLengthCm (se for com o termo independente, possui uma coluna a mais que só contém valores 1).

# 2 Questão 1

## 2.1 Resultados

### 2.1.1 Iris-Setosa

```
1 Iris-Setosa
2
3 SEM O TERMO INDEPENDENTE:
4 y = a*x1 + b*x2 + c*x3
5 [a b c] = [ 0.07455282 -0.06602361 0.03673264]
```

```

6
7 COM O TERMO INDEPENDENTE:
8 y = a*x1 + b*x2 + c*x3 + k
9 [a b c k] = [ 0.13497209 -0.07886596  0.10365958 -0.36144997]

```

### 2.1.2 Iris-Versicolor

```

1 Iris-Versicolor
2
3 SEM O TERMO INDEPENDENTE:
4 y = a*x1 + b*x2 + c*x3
5 [a b c] = [-0.14382683  0.17051618  0.40397714]
6
7 COM O TERMO INDEPENDENTE:
8 y = a*x1 + b*x2 + c*x3 + k
9 [a b c k] = [-0.06826027  0.24149193  0.39877688 -0.63863516]

```

### 2.1.3 Iris-Virginica

```

1 Iris-Virginica
2
3 SEM O TERMO INDEPENDENTE:
4 y = a*x1 + b*x2 + c*x3
5 [a b c] = [-0.11329721  0.3990124  0.25976859]
6
7 COM O TERMO INDEPENDENTE:
8 y = a*x1 + b*x2 + c*x3 + k
9 [a b c k] = [-0.11686468  0.35346044  0.22325742  0.36734017]

```

## 3 Questão 2

### 3.1 Resultados

#### 3.1.1 Iris-Setosa

```

1 Iris-Setosa
2
3 R = VΛV^(T)
4
5 SEM O TERMO INDEPENDENTE:
6 V = [[-0.80618188 -0.45661854  0.37625828]
7      [-0.54157827  0.31342349 -0.78003762]
8      [-0.23825146  0.8326255  0.49997102]]
9
10 Λ = [[5.86392634e+02  0.00000000e+00  0.00000000e+00]
11      [0.00000000e+00  5.29776824e-01  0.00000000e+00]
12      [0.00000000e+00  0.00000000e+00  7.47589571e-01]]
13
14 V^T = [[-0.80618188 -0.54157827 -0.23825146]
15         [-0.45661854  0.31342349  0.8326255 ]

```

```

16 [ 0.37625828 -0.78003762 0.49997102]]
17
18
19 COM O TERMO INDEPENDENTE:
20 V = [[-0.79610972 0.1628056 -0.47792329 -0.33360603]
21 [-0.53478061 -0.03864172 0.33762985 0.77364243]
22 [-0.23529777 0.18572124 0.80860497 -0.50626137]
23 [-0.15765144 -0.96825037 0.06126507 -0.18407563]]
24
25 Λ = [[6.01336860e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00]
26 [0.00000000e+00 2.79352229e-02 0.00000000e+00 0.00000000e+00]
27 [0.00000000e+00 0.00000000e+00 5.31947766e-01 0.00000000e+00]
28 [0.00000000e+00 0.00000000e+00 0.00000000e+00 7.73257486e-01]]
29
30 V^T = [[-0.79610972 -0.53478061 -0.23529777 -0.15765144]
31 [ 0.1628056 -0.03864172 0.18572124 -0.96825037]
32 [-0.47792329 0.33762985 0.80860497 0.06126507]
33 [-0.33360603 0.77364243 -0.50626137 -0.18407563]]

```

### 3.1.2 Iris-Versicolor

```

1 Iris-Versicolor
2
3 R = VΛV^T
4
5 SEM O TERMO INDEPENDENTE:
6 V = [[ 0.76039163 0.64759888 -0.0491961 ]
7 [ 0.35223975 -0.47485674 -0.80649751]
8 [ 0.54564799 -0.59592513 0.58918716]]
9
10 Λ = [[9.59570396e+02 0.00000000e+00 0.00000000e+00]
11 [0.00000000e+00 1.33162776e+00 0.00000000e+00]
12 [0.00000000e+00 0.00000000e+00 9.57976568e-01]]
13
14 V^T = [[ 0.76039163 0.35223975 0.54564799]
15 [ 0.64759888 -0.47485674 -0.59592513]
16 [-0.0491961 -0.80649751 0.58918716]]
17
18
19 COM O TERMO INDEPENDENTE:
20 V = [[-0.7545525 0.11812875 0.64474223 0.03167953]
21 [-0.34954066 0.11526607 -0.46962259 0.80248968]
22 [-0.54144512 -0.01367345 -0.60234554 -0.58637025]
23 [-0.1237297 -0.98619084 0.030691 0.1057197 ]]
24
25 Λ = [[9.74487597e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00]
26 [0.00000000e+00 7.14132919e-02 0.00000000e+00 0.00000000e+00]
27 [0.00000000e+00 0.00000000e+00 1.33280153e+00 0.00000000e+00]
28 [0.00000000e+00 0.00000000e+00 0.00000000e+00 9.68188359e-01]]
29
30 V^T = [[-0.7545525 -0.34954066 -0.54144512 -0.1237297 ]
31 [ 0.11812875 0.11526607 -0.01367345 -0.98619084]

```

```

32 [ 0.64474223 -0.46962259 -0.60234554 0.030691 ]
33 [ 0.03167953 0.80248968 -0.58637025 0.1057197 ]]

```

### 3.1.3 Iris-Virginica

```

1 Iris-Virginica
2
3 R = VΛV^(T)
4
5 SEM O TERMO INDEPENDENTE:
6 V = [[-0.72592866 -0.63483621 0.2645951 ]
7       [-0.32772296 -0.01894783 -0.94458385]
8       [-0.60466953 0.77241438 0.19429563]]
9
10 Λ = [[1.28249881e+03 0.00000000e+00 0.00000000e+00]
11       [0.00000000e+00 5.79253032e-01 0.00000000e+00]
12       [0.00000000e+00 0.00000000e+00 1.07193335e+00]]
13
14 V^T = [[-0.72592866 -0.32772296 -0.60466953]
15         [-0.63483621 -0.01894783 0.77241438]
16         [0.2645951 -0.94458385 0.19429563]]
17
18
19 COM O TERMO INDEPENDENTE:
20 V = [[-7.21743001e-01 -2.76297507e-01 6.34623278e-01 -1.50933108e-04]
21       [-3.25843186e-01 9.35769444e-01 3.68031853e-02 -1.29642937e-01]
22       [-6.01187515e-01 -1.93722166e-01 -7.68083597e-01 -1.05322749e-01]
23       [-1.07176625e-01 1.02308151e-01 -7.71129602e-02 9.85951218e-01]]
24
25 Λ = [[1.29740071e+03 0.00000000e+00 0.00000000e+00 0.00000000e+00]
26       [0.00000000e+00 1.08243546e+00 0.00000000e+00 0.00000000e+00]
27       [0.00000000e+00 0.00000000e+00 5.82311582e-01 0.00000000e+00]
28       [0.00000000e+00 0.00000000e+00 0.00000000e+00 8.45463191e-02]]
29
30 V^T = [[-7.21743001e-01 -3.25843186e-01 -6.01187515e-01 -1.07176625e-01]
31         [-2.76297507e-01 9.35769444e-01 -1.93722166e-01 1.02308151e-01]
32         [6.34623278e-01 3.68031853e-02 -7.68083597e-01 -7.71129602e-02]
33         [-1.50933108e-04 -1.29642937e-01 -1.05322749e-01 9.85951218e-01]]

```

## 4 Questão 3

### 4.1 Resultados

#### 4.1.1 Iris-Setosa

```

1 Iris-Setosa
2
3 R = UΣV^(T)
4
5 SEM O TERMO INDEPENDENTE:
6 U = [[-0.80618188 0.37625828 -0.45661854]

```

```

7  [-0.54157827 -0.78003762  0.31342349]
8  [-0.23825146  0.49997102  0.8326255  ]]
9
10 Σ =  [[5.86392634e+02  0.00000000e+00  0.00000000e+00]
11        [0.00000000e+00  7.47589571e-01  0.00000000e+00]
12        [0.00000000e+00  0.00000000e+00  5.29776824e-01]]
13
14 V^T =  [[-0.80618188 -0.54157827 -0.23825146]
15          [ 0.37625828 -0.78003762  0.49997102]
16          [-0.45661854  0.31342349  0.8326255  ]]
17
18
19 COM O TERMO INDEPENDENTE:
20 U =  [[-0.79610972  0.33360603  0.47792329 -0.1628056  ]
21        [-0.53478061 -0.77364243 -0.33762985  0.03864172]
22        [-0.23529777  0.50626137 -0.80860497 -0.18572124]
23        [-0.15765144  0.18407563 -0.06126507  0.96825037]]
24
25 Σ =  [[6.01336860e+02  0.00000000e+00  0.00000000e+00  0.00000000e+00]
26        [0.00000000e+00  7.73257486e-01  0.00000000e+00  0.00000000e+00]
27        [0.00000000e+00  0.00000000e+00  5.31947766e-01  0.00000000e+00]
28        [0.00000000e+00  0.00000000e+00  0.00000000e+00  2.79352229e-02]]
29
30 V^T =  [[-0.79610972 -0.53478061 -0.23529777 -0.15765144]
31          [ 0.33360603 -0.77364243  0.50626137  0.18407563]
32          [ 0.47792329 -0.33762985 -0.80860497 -0.06126507]
33          [-0.1628056  0.03864172 -0.18572124  0.96825037]]

```

#### 4.1.2 Iris-Versicolor

```

1  Iris-Versicolor
2
3  R = UΣV^T
4
5  SEM O TERMO INDEPENDENTE:
6  U =  [[-0.76039163  0.64759888 -0.0491961  ]
7        [-0.35223975 -0.47485674 -0.80649751]
8        [-0.54564799 -0.59592513  0.58918716]]
9
10 Σ =  [[9.59570396e+02  0.00000000e+00  0.00000000e+00]
11        [0.00000000e+00  1.33162776e+00  0.00000000e+00]
12        [0.00000000e+00  0.00000000e+00  9.57976568e-01]]
13
14 V^T =  [[-0.76039163 -0.35223975 -0.54564799]
15          [ 0.64759888 -0.47485674 -0.59592513]
16          [-0.0491961  -0.80649751  0.58918716]]
17
18
19 COM O TERMO INDEPENDENTE:
20 U =  [[-0.7545525  0.64474223  0.03167953 -0.11812875]
21        [-0.34954066 -0.46962259  0.80248968 -0.11526607]
22        [-0.54144512 -0.60234554 -0.58637025  0.01367345]

```

```

23 [-0.1237297 0.030691 0.1057197 0.98619084]]
24
25 Σ = [[9.74487597e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00]
26 [0.00000000e+00 1.33280153e+00 0.00000000e+00 0.00000000e+00]
27 [0.00000000e+00 0.00000000e+00 9.68188359e-01 0.00000000e+00]
28 [0.00000000e+00 0.00000000e+00 0.00000000e+00 7.14132919e-02]]
29
30 V^T = [[-0.7545525 -0.34954066 -0.54144512 -0.1237297 ]
31 [ 0.64474223 -0.46962259 -0.60234554 0.030691 ]
32 [ 0.03167953 0.80248968 -0.58637025 0.1057197 ]
33 [-0.11812875 -0.11526607 0.01367345 0.98619084]]

```

### 4.1.3 Iris-Virginica

```

1 Iris-Virginica
2
3 R = UΣV^T
4
5 SEM O TERMO INDEPENDENTE:
6 U = [[-0.72592866 0.2645951 -0.63483621]
7 [-0.32772296 -0.94458385 -0.01894783]
8 [-0.60466953 0.19429563 0.77241438]]
9
10 Σ = [[1.28249881e+03 0.00000000e+00 0.00000000e+00]
11 [0.00000000e+00 1.07193335e+00 0.00000000e+00]
12 [0.00000000e+00 0.00000000e+00 5.79253032e-01]]
13
14 V^T = [[-0.72592866 -0.32772296 -0.60466953]
15 [ 0.2645951 -0.94458385 0.19429563]
16 [-0.63483621 -0.01894783 0.77241438]]
17
18
19 COM O TERMO INDEPENDENTE:
20 U = [[-7.21743001e-01 2.76297507e-01 6.34623278e-01 -1.50933108e-04]
21 [-3.25843186e-01 -9.35769444e-01 3.68031853e-02 -1.29642937e-01]
22 [-6.01187515e-01 1.93722166e-01 -7.68083597e-01 -1.05322749e-01]
23 [-1.07176625e-01 -1.02308151e-01 -7.71129602e-02 9.85951218e-01]]
24
25 Σ = [[1.29740071e+03 0.00000000e+00 0.00000000e+00 0.00000000e+00]
26 [0.00000000e+00 1.08243546e+00 0.00000000e+00 0.00000000e+00]
27 [0.00000000e+00 0.00000000e+00 5.82311582e-01 0.00000000e+00]
28 [0.00000000e+00 0.00000000e+00 0.00000000e+00 8.45463191e-02]]
29
30 V^T = [[-7.21743001e-01 -3.25843186e-01 -6.01187515e-01 -1.07176625e-01]
31 [ 2.76297507e-01 -9.35769444e-01 1.93722166e-01 -1.02308151e-01]
32 [ 6.34623278e-01 3.68031853e-02 -7.68083597e-01 -7.71129602e-02]
33 [-1.50933108e-04 -1.29642937e-01 -1.05322749e-01 9.85951218e-01]]

```



## 5 Questão 4

### 5.1 Resultados

```
1 SEM O TERMO INDEPENDENTE:
2 A = Iris-Versicolor
3 B = Iris-Setosa
4 C = Iris-Setosa
5 D = Iris-Versicolor
6 E = Iris-Virginica
7
8 COM O TERMO INDEPENDENTE:
9 A = Iris-Versicolor
10 B = Iris-Setosa
11 C = Iris-Setosa
12 D = Iris-Versicolor
13 E = Iris-Virginica
```

## 6 Código

```
1 # Programa codigo.py
2 # Autora: Karen dos Anjos Arcoverde
3 # Data: 06/02/2021
4 #
5
6
7 import numpy as np
8 from scipy import linalg
9 from scipy.linalg import diagsvd
10 import sys
11
12
13
14 ##### Funcoes #####
15 def pegarDados(tipo_iris):
16
17     # tipo_iris = 1 Setosa , tipo_iris = 2 Versicolor,
18     # tipo_iris = 3 Virginica
19     dados = []
20     IDs = []
21
22     Setosa = range(25,39+1)
23     Versicolor = range(75,89+1)
24     Virginica = range (125,139+1)
25
26     arquivo= open("dados_13.csv",'r')
27     arquivo.readline() # ignora a primeira linha
28
29     if (tipo_iris == 1):
30         IDs = Setosa
31     if (tipo_iris == 2):
```

```

32     IDs = Versicolor
33     if (tipo_iris == 3):
34         IDs = Virginica
35
36     for i in range(1,46): # percorre todo o banco de dados 1-45
37         linha = (arquivo.readline()).split(',') #separa os dados por virgula
38
39
40         if (int(linha[0]) in IDs):# percorre os ids selecionados
41             linha.pop(0) # retira o ID dos dados
42             linha.pop(-1) # retira a especie dos dados
43             for j in range(4): #para cada dado
44                 linha[j] = float(linha[j]) #transforma em numero
45
46             dados.append(linha) #adiciona na lista de dados
47
48     arquivo.close()
49     return dados
50
51 # -----
52 def construir_equacao_normal (dados):
53
54     #equacao normal - minimos quadrados:
55     #  $(x^T) \cdot x \cdot w = (x^T) \cdot y$ 
56     #  $(x\_transposta) \cdot x \cdot w = (x\_transposta) \cdot y$ 
57     #  $R \cdot w = p$ 
58     #  $R = (x^T) \cdot x$ 
59     #  $p = (x^T) \cdot y$ 
60
61     ##### sem termo independente #####
62     #  $y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3$ 
63     x = []
64     y = []
65
66     #achar  $(x^T)$  e x
67     # x - colunas SepalLengthCm, SepalWidthCm, PetalLengthCm
68     x = np.array(dados)
69     x = np.delete(x.reshape(15,4),3,1) #deleta a ultima coluna de x
70     x_transposta = np.transpose(x) #faz a transposta de x:  $(x^T)$ 
71
72     #achar R
73     R = np.dot(x_transposta,x) #multiplica  $(x^T)$  por x
74
75     #achar y - coluna PetalWidthCm
76     y = np.array(dados)
77     y = np.delete(y.reshape(15,4),0,1) #deleta a primeira coluna de y
78     y = np.delete(y.reshape(15,3),0,1) #deleta a segunda coluna de y
79     y = np.delete(y.reshape(15,2),0,1) #deleta a terceira coluna de y
80
81     #achar p
82     p = np.dot(x_transposta,y)
83

```

```

84 ##### com termo independente #####
85 # y = a*x1 +b*x2 +c*x3 +k
86
87 for i in range (0,15):
88     dados[i][3] = 1
89
90 #achar (x^T) e x
91 x = np.array(dados)
92 x_transposta = np.transpose(x) #faz a transposta de x: (x^T)
93 #achar R
94 R1= np.dot(x_transposta,x) #multiplica (x^T) por x
95 #achar p
96 p1 = np.dot(x_transposta,y)
97
98 return R,p,R1,p1
99
100 # -----
101 def PLU(R,p):
102     indice_coluna = 0
103     indice_linha = 1
104     indice_L_1s = 0
105     tamanho_R = len(R)
106
107     while (indice_coluna < tamanho_R):
108         #constroi a matriz L, triangular inferior
109         if (R[indice_coluna][indice_coluna] != 0):
110             L = np.zeros((tamanho_R, tamanho_R))
111
112             indice_coluna_aux = indice_coluna + 1
113             while (indice_coluna_aux < tamanho_R):
114                 L[indice_coluna_aux][indice_coluna] = - R[indice_coluna_aux][
115                     indice_coluna]/R[indice_coluna][indice_coluna]
116                 indice_linha += 1
117                 indice_coluna_aux += 1
118
119             while (indice_L_1s < tamanho_R):
120                 L[indice_L_1s][indice_L_1s] = 1
121                 indice_L_1s += 1
122
123             indice_L_1s = 0
124             indice_linha = 1
125
126             #multiplica a matriz R por L
127             R = np.dot (L,R)
128             p = np.dot (L,p)
129
130             #se o pivo for zero, necessario multiplicar por uma matriz P de
131             permutacao
132             if (R[indice_coluna][indice_coluna] == 0):
133                 P = np.zeros((tamanho_R, tamanho_R))
134                 P[indice_coluna][indice_coluna + 1] = 1

```

```

134         P[indice_coluna + 1][indice_coluna] = 1
135
136         i = 0
137         j = 0
138         guarda_1 = False
139         while (i < tamanho_R):
140             while (j < tamanho_R):
141                 if (P[i][j] == 1):
142                     guarda_1 = True
143                     j += 1
144                 if (guarda_1 == False):
145                     P[i][i] = 1
146                 i += 1
147
148         R = np.dot(P,R)
149         p = np.dot (P,p)
150
151         indice_coluna += 1
152
153     #backsubstitution
154     w = np.linalg.solve(R, p)
155
156     return w
157
158 # -----
159 def decomposicao_espectral(R):
160
161     #  $R = VDV^T$ 
162     #determinando autovalores e autovetores
163     autovalores, autovetores = np.linalg.eig(R)
164     # matriz diagonal de autovalores
165     matrizDiagonal = np.diag(autovalores)
166
167     return autovetores, matrizDiagonal
168
169 # -----
170 def estimar_amstras(amostra, w_setosa,w_versicolor,w_virginica,c_independente
171 ):
172
173     lista_erros = []
174     estimativa = ""
175     indice = 0
176     amostra_x = []
177     indice_erros_modulo = 0
178
179     while (indice < 3):
180         amostra_x.append (amostra[indice])
181         indice += 1
182
183     if (c_independente == True):
184         amostra_x.append(1)

```

```

185 amostra_x = np.array(amostra_x)
186
187 #produto interno <x,y> = (x^T).y
188
189 estimativa_setosa = np.dot(amostra_x,w_setosa)
190 estimativa_versicolor = np.dot(amostra_x,w_versicolor)
191 estimativa_virginica = np.dot(amostra_x,w_virginica)
192
193 erro_setosa = estimativa_setosa[0] - amostra[3]
194 lista_erros.append(erro_setosa)
195
196 erro_versicolor = estimativa_versicolor[0] - amostra[3]
197 lista_erros.append(erro_versicolor)
198
199 erro_virginica = estimativa_virginica[0] - amostra[3]
200 lista_erros.append(erro_virginica)
201
202 while (indice_erros_modulo < len(lista_erros)):
203     lista_erros[indice_erros_modulo] = abs(lista_erros[indice_erros_modulo
204         ])
205     indice_erros_modulo += 1
206
207 if (min(lista_erros) == abs(erro_setosa)):
208     estimativa = "Iris-Setosa"
209
210     return estimativa
211
212 if (min(lista_erros) == abs(erro_versicolor)):
213     estimativa = "Iris-Versicolor"
214
215     return estimativa
216
217 if (min(lista_erros) == abs(erro_virginica)):
218     estimativa = "Iris-Virginica"
219
220     return estimativa
221
222 ##### Programa Principal #####
223 def menu():
224     resultado = 0
225     tipo_iris = 0
226     coeficientes_sem_aux = []
227     coeficientes_com_aux = []
228     indice = 0
229
230     while (resultado != 5):
231         print()
232         print('##### MENU PRINCIPAL
233             #####')
234         print("Digite somente o numero da questao que voce deseja ver o
235             resultado: ")

```

```

234 print("1 = Questao 1")
235 print("2 = Questao 2")
236 print("3 = Questao 3")
237 print("4 = Questao 4")
238 print("5 = SAIR")
239 print('
    #####
    ')
240 print()
241
242 resultado = int(input())
243
244 if (resultado == 5):
245     sys.exit(0)
246
247 if (resultado == 1):
248
249     while (tipo_iris != 4):
250         print('##### MENU IRIS
251             #####')
252         print("Digite qual especie voce deseja fazer a regressao
253             linear: ")
254         print("1 = Iris-Setosa")
255         print("2 = Iris-Versicolor")
256         print("3 = Iris Virginica")
257         print("4 = VOLTAR AO MENU PRINCIPAL")
258         print('
259             #####')
260         print()
261
262         tipo_iris = int(input())
263
264         if (tipo_iris == 4):
265             menu()
266
267         else:
268             dados = pegarDados (tipo_iris)
269             R,p,R1,p1 = construir_equacao_normal(dados)
270             w = PLU(R,p)
271             w1 = PLU(R1,p1)
272
273             print()
274             if (tipo_iris == 1):
275                 print("Iris-Setosa\n")
276             elif (tipo_iris == 2):
277                 print("Iris-Versicolor\n")
278             elif (tipo_iris == 3):
279                 print("Iris-Virginica\n")
280
281             print("SEM O TERMO INDEPENDENTE: ")
282             print("y = a*x1 + b*x2 + c*x3")
283             print("[a b c] = ",end="")

```

```

281
282
283     while (indice < len(w)):
284         coeficientes_sem_aux.append(w[indice][0])
285         coeficientes_sem = np.array(coeficientes_sem_aux)
286         indice += 1
287
288     print(coeficientes_sem)
289     coeficientes_sem_aux = []
290
291
292     print()
293
294     print("COM O TERMO INDEPENDENTE: ")
295     print("y = a*x1 + b*x2 + c*x3 + k")
296     print("[a b c k] = ",end="")
297
298     indice = 0
299     while (indice < len(w1)):
300         coeficientes_com_aux.append(w1[indice][0])
301         coeficientes_com = np.array(coeficientes_com_aux)
302         indice += 1
303
304     print(coeficientes_com)
305     coeficientes_com_aux = []
306     indice = 0
307
308     print()
309
310 if (resultado == 2):
311
312     while (tipo_iris != 4):
313         print('##### MENU IRIS
314             #####')
315         print("Digite qual especie voce deseja fazer a decomposicao
316             espectral: ")
317         print("1 = Iris-Setosa")
318         print("2 = Iris-Versicolor")
319         print("3 = Iris Virginica")
320         print("4 = VOLTAR AO MENU PRINCIPAL")
321         print('
322             #####
323             ')
324         print()
325
326         tipo_iris = int(input())
327
328         if (tipo_iris == 4):
329             menu()
330
331         else:

```

```

329     dados = pegarDados (tipo_iris)
330     R,p,R1,p1 = construir_equacao_normal(dados)
331     autovetores, matrizDiagonal = decomposicao_espectral(R)
332     autovetores1, matrizDiagonal1 = decomposicao_espectral(R1)
333
334     if (tipo_iris == 1):
335         print("Iris-Setosa\n")
336     elif (tipo_iris == 2):
337         print("Iris-Versicolor\n")
338     elif (tipo_iris == 3):
339         print("Iris-Virginica\n")
340
341     print("R = V\u039BV^(T)")
342     print()
343
344     print("SEM O TERMO INDEPENDENTE: ")
345     print("V = ",autovetores)
346     print()
347     print('\u039B = ',matrizDiagonal)
348     print()
349     print("V^T = ",np.transpose(autovetores))
350
351     print()
352     print()
353
354     print("COM O TERMO INDEPENDENTE: ")
355     print("V = ", autovetores1)
356     print()
357     print('\u039B = ', matrizDiagonal1)
358     print()
359     print("V^T = ", np.transpose(autovetores1))
360
361     print()
362
363 if (resultado == 3):
364
365     while (tipo_iris != 4):
366         print('##### MENU IRIS #####')
367         print("Digite qual especie voce deseja fazer o SVD: ")
368         print("1 = Iris-Setosa")
369         print("2 = Iris-Versicolor")
370         print("3 = Iris Virginica")
371         print("4 = VOLTAR AO MENU PRINCIPAL")
372         print('#####')
373         print()
374
375         tipo_iris = int(input())
376
377     if (tipo_iris == 4):
378         menu()
379
380 else:

```



```

381
382     dados = pegarDados (tipo_iris)
383     R,p,R1,p1 = construir_equacao_normal(dados)
384     U, s, VT = linalg.svd(R)
385     U1, s1, VT1 = linalg.svd(R1)
386
387     if (tipo_iris == 1):
388         print("Iris-Setosa\n")
389     elif (tipo_iris == 2):
390         print("Iris-Versicolor\n")
391     elif (tipo_iris == 3):
392         print("Iris-Virginica\n")
393
394     print("R = U\u03A3V^(T)")
395     print()
396
397     print("SEM O TERMO INDEPENDENTE: ")
398     print("U = ", U)
399     print()
400     print('\u03A3 = ', diagsvd(s, R.shape[0], R.shape[1]))
401     print()
402     print("V^T = ", VT)
403     print()
404     print()
405
406     print("COM O TERMO INDEPENDENTE: ")
407     print("U = ", U1)
408     print()
409     print('\u03A3 = ', diagsvd(s1, R1.shape[0], R1.shape[1]))
410     print()
411     print("V^T = ", VT1)
412
413     print()
414
415
416
417 if (resultado == 4):
418     A = [5.0,2.3,3.3,1.0]
419     B = [4.6,3.2,1.4,0.2]
420     C = [5.0,3.3,1.4,0.2]
421     D = [6.1,3.0,4.6,1.4]
422     E = [5.9,3.0,5.1,1.8]
423
424     print("SEM O TERMO INDEPENDENTE: ")
425     dados = pegarDados (1)
426     R,p,R1,p1 = construir_equacao_normal(dados)
427     w_setosa = PLU(R,p)
428
429     dados = pegarDados (2)
430     R,p,R1,p1 = construir_equacao_normal(dados)
431     w_versicolor = PLU(R,p)
432

```

```

433 dados = pegarDados (3)
434 R,p,R1,p1 = construir_equacao_normal(dados)
435 w_virginica = PLU(R,p)
436
437 c_independente = False
438 estimativa = estimar_amstras(A,w_setosa,w_versicolor,w_virginica,
439                               c_independente)
439 print("A = ", estimativa)
440 estimativa = estimar_amstras(B,w_setosa,w_versicolor,w_virginica,
441                               c_independente)
441 print("B = ", estimativa)
442 estimativa = estimar_amstras(C,w_setosa,w_versicolor,w_virginica,
443                               c_independente)
443 print("C = ", estimativa)
444 estimativa = estimar_amstras(D,w_setosa,w_versicolor,w_virginica,
445                               c_independente)
445 print("D = ", estimativa)
446 estimativa = estimar_amstras(E,w_setosa,w_versicolor,w_virginica,
447                               c_independente)
447 print("E = ", estimativa)
448
449 print()
450 print("COM O TERMO INDEPENDENTE: ")
451 dados = pegarDados (1)
452 R,p,R1,p1 = construir_equacao_normal(dados)
453 w1_setosa = PLU(R1,p1)
454
455 dados = pegarDados (2)
456 R,p,R1,p1 = construir_equacao_normal(dados)
457 w1_versicolor = PLU(R1,p1)
458
459 dados = pegarDados (3)
460 R,p,R1,p1 = construir_equacao_normal(dados)
461 w1_virginica = PLU(R1,p1)
462
463 c_independente = True
464 estimativa1 = estimar_amstras(A,w1_setosa,w1_versicolor,
465                               w1_virginica,c_independente)
465 print("A = ", estimativa1)
466 estimativa1 = estimar_amstras(B,w1_setosa,w1_versicolor,
467                               w1_virginica,c_independente)
467 print("B = ", estimativa1)
468 estimativa1 = estimar_amstras(C,w1_setosa,w1_versicolor,
469                               w1_virginica,c_independente)
469 print("C = ", estimativa1)
470 estimativa1 = estimar_amstras(D,w1_setosa,w1_versicolor,
471                               w1_virginica,c_independente)
471 print("D = ", estimativa1)
472 estimativa1 = estimar_amstras(E,w1_setosa,w1_versicolor,
473                               w1_virginica,c_independente)
473 print("E = ", estimativa1)
474

```

```
475         menu()
```

```
476  
477 ##### chamada ao menu  
478 menu()
```

## 7 Bibliografia

- <https://algebralinearufcg.github.io/jup-not/prog02-learning-numpy.html>
- <https://machinelearningmastery.com/singular-value-decomposition-for-machine-learning/>
- <https://pt.coredump.biz/questions/34007632/how-to-remove-a-column-in-a-numpy-array>
- <https://pythonforundergradengineers.com/unicode-characters-in-python.html>
- <https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.svd.html>
- <https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.diagsvd.html>
- <https://www.geeksforgeeks.org/python-exit-commands-quit-exit-sys-exit-and-os-exit/>