

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>18</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 = V\V^(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
11
12 ##### Funcoes #####
13 def pegarDados(tipo_iris):
14
15     # tipo_iris = 1 Setosa , tipo_iris = 2 Versicolor,
16     # tipo_iris = 3 Virginica
17     dados = []
18     IDs = []
19
20     Setosa = range(25,39+1)
21     Versicolor = range(75,89+1)
22     Virginica = range (125,139+1)
23
24     arquivo= open("dados_13.csv",'r')
25     arquivo.readline() # ignora a primeira linha

```

```

26
27     if (tipo_iris == '1'):
28         IDs = Setosa
29     if (tipo_iris == '2'):
30         IDs = Versicolor
31     if (tipo_iris == '3'):
32         IDs = Virginica
33
34     for i in range(1,46): # percorre todo o banco de dados 1-45
35         linha = (arquivo.readline()).split(',') #separa os dados por
            virgula
36
37
38         if (int(linha[0]) in IDs):# percorre os ids selecionados
39             linha.pop(0) # retira o ID dos dados
40             linha.pop(-1) # retira a especie dos dados
41             for j in range(4): #para cada dado
42                 linha[j] = float(linha[j]) #transforma em numero
43
44             dados.append(linha) #adiciona na lista de dados
45
46     arquivo.close()
47     return dados
48
49 # -----
50 def construir_equacao_normal (dados):
51
52     #equacao normal - minimos quadrados:
53     #  $(x^T) \cdot x \cdot w = (x^T) \cdot y$ 
54     #  $(x\_transposta) \cdot x \cdot w = (x\_transposta) \cdot y$ 
55     #  $R \cdot w = p$ 
56     #  $R = (x^T) \cdot x$ 
57     #  $p = (x^T) \cdot y$ 
58
59     ##### sem termo independente #####
60     #  $y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3$ 
61     x = []
62     y = []
63
64     #achar  $(x^T)$  e x
65     # x - colunas SepalLengthCm, SepalWidthCm, PetalLengthCm
66     x = np.array(dados)
67     x = np.delete(x.reshape(15,4),3,1) #deleta a ultima coluna de x
68     x_transposta = np.transpose(x) #faz a transposta de x:  $(x^T)$ 
69
70     #achar R
71     R = np.dot(x_transposta,x) #multiplica  $(x^T)$  por x
72
73     #achar y - coluna PetalWidthCm
74     y = np.array(dados)
75     y = np.delete(y.reshape(15,4),0,1) #deleta a primeira coluna de y

```

```

76 y = np.delete(y.reshape(15,3),0,1) #deleta a segunda coluna de y
77 y = np.delete(y.reshape(15,2),0,1) #deleta a terceira coluna de y
78
79 #achar p
80 p = np.dot(x_transposta,y)
81
82 ##### com termo independente #####
83 # y = a*x1 +b*x2 +c*x3 +k
84
85 for i in range (0,15):
86     dados[i][3] = 1
87
88 #achar (x^T) e x
89 x = np.array(dados)
90 x_transposta = np.transpose(x) #faz a transposta de x: (x^T)
91 #achar R
92 R1= np.dot(x_transposta,x) #multiplica (x^T) por x
93 #achar p
94 p1 = np.dot(x_transposta,y)
95
96 return R,p,R1,p1
97
98 # -----
99 def PLU(R,p):
100     indice_coluna = 0
101     indice_linha = 1
102     indice_L_1s = 0
103     tamanho_R = len(R)
104
105     while (indice_coluna < tamanho_R):
106         #constroi a matriz L, triangular inferior
107         if (R[indice_coluna][indice_coluna] != 0):
108             L = np.zeros((tamanho_R, tamanho_R))
109
110             indice_coluna_aux = indice_coluna + 1
111             while (indice_coluna_aux < tamanho_R):
112                 L[indice_coluna_aux][indice_coluna] = - R[
113                     indice_coluna_aux][indice_coluna]/R[indice_coluna][
114                         indice_coluna]
115                 indice_linha += 1
116                 indice_coluna_aux += 1
117
118             while (indice_L_1s < tamanho_R):
119                 L[indice_L_1s][indice_L_1s] = 1
120                 indice_L_1s += 1
121
122             indice_L_1s = 0
123             indice_linha = 1
124
125         #multiplica a matriz R por L
126         R = np.dot (L,R)

```

```

125         p = np.dot (L,p)
126
127
128         #se o pivo for zero, necessario multiplicar por uma matriz P de
        permutacao
129         if (R[indice_coluna][indice_coluna] == 0):
130             P = np.zeros((tamanho_R, tamanho_R))
131             P[indice_coluna][indice_coluna + 1] = 1
132             P[indice_coluna + 1][indice_coluna] = 1
133
134             i = 0
135             j = 0
136             guarda_1 = False
137             while (i < tamanho_R):
138                 while (j < tamanho_R):
139                     if (P[i][j] == 1):
140                         guarda_1 = True
141                         j += 1
142                     if (guarda_1 == False):
143                         P[i][i] = 1
144                     i += 1
145
146             R = np.dot(P,R)
147             p = np.dot (P,p)
148
149             indice_coluna += 1
150
151         #backsubstitution
152         w = np.linalg.solve(R, p)
153
154         return w
155
156 # -----
157 def decomposicao_espectral(R):
158
159     #  $R = VDV^T$ 
160     #determinando autovalores e autovetores
161     autovalores, autovetores = np.linalg.eig(R)
162     # matriz diagonal de autovalores
163     matrizDiagonal = np.diag(autovalores)
164
165     return autovetores, matrizDiagonal
166
167 # -----
168 def estimar_amostras(amostra, w_setosa,w_versicolor,w_virginica,
        c_independente):
169
170     lista_erros = []
171     estimativa = ""
172     indice = 0
173     amostra_x = []

```

```

174     indice_erros_modulo = 0
175
176     while (indice < 3):
177         amostra_x.append (amostra[indice])
178         indice += 1
179
180     if (c_independente == True):
181         amostra_x.append(1)
182
183     amostra_x = np.array(amostra_x)
184
185     #produto interno <x,y> = (x^T).y
186
187     estimativa_setosa = np.dot(amostra_x,w_setosa)
188     estimativa_versicolor = np.dot(amostra_x,w_versicolor)
189     estimativa_virginica = np.dot(amostra_x,w_virginica)
190
191     erro_setosa = estimativa_setosa[0] - amostra[3]
192     lista_erros.append(erro_setosa)
193
194     erro_versicolor = estimativa_versicolor[0] - amostra[3]
195     lista_erros.append(erro_versicolor)
196
197     erro_virginica = estimativa_virginica[0] - amostra[3]
198     lista_erros.append(erro_virginica)
199
200     while (indice_erros_modulo < len(lista_erros)):
201         lista_erros [indice_erros_modulo] = abs(lista_erros[
202             indice_erros_modulo])
203         indice_erros_modulo += 1
204
205     if (min(lista_erros) == abs(erro_setosa)):
206         estimativa = "Iris-Setosa"
207
208         return estimativa
209
210     if (min(lista_erros) == abs(erro_versicolor)):
211         estimativa = "Iris-Versicolor"
212
213         return estimativa
214
215     if (min(lista_erros) == abs(erro_virginica)):
216         estimativa = "Iris-Virginica"
217
218         return estimativa
219
220 ##### Programa Principal #####
221 def menu():
222     resultado = ""
223     tipo_iris = ""
224     coeficientes_sem_aux = []

```

```

224     coeficientes_com_aux = []
225     indice = 0
226
227     print()
228     print("Digite somente o numero da questao que voce deseja ver o
          resultado: ")
229     print("Questao 1")
230     print("Questao 2")
231     print("Questao 3")
232     print("Questao 4")
233     print()
234     print("PARA SAIR DIGITE 0")
235
236     while (resultado != '0'):
237         resultado = input()
238
239         if (resultado == '1'):
240             print("Digite qual especie voce deseja fazer a regressao
                  linear: ")
241             print("1 = Iris-Setosa")
242             print("2 = Iris-Versicolor")
243             print("3 = Iris Virginica")
244             print()
245             print("PARA SAIR DIGITE 0")
246
247             while (tipo_iris != '0'):
248
249                 tipo_iris = input()
250
251                 if (tipo_iris == '0'):
252                     break
253
254                 dados = pegarDados (tipo_iris)
255                 R,p,R1,p1 = construir_equacao_normal(dados)
256                 w = PLU(R,p)
257                 w1 = PLU(R1,p1)
258
259                 print()
260                 if (tipo_iris == '1'):
261                     print("Iris-Setosa\n")
262                 elif (tipo_iris == '2'):
263                     print("Iris-Versicolor\n")
264                 elif (tipo_iris == '3'):
265                     print("Iris-Virginica\n")
266
267                 print("SEM O TERMO INDEPENDENTE: ")
268                 print("y = a*x1 + b*x2 + c*x3")
269                 print("[a b c] = ",end="")
270
271
272         while (indice < len(w)):

```

```

273         coeficientes_sem_aux.append(w[indice][0])
274         coeficientes_sem = np.array(coeficientes_sem_aux)
275         indice += 1
276
277     print(coeficientes_sem)
278     coeficientes_sem_aux = []
279
280
281     print()
282
283     print("COM O TERMO INDEPENDENTE: ")
284     print("y = a*x1 + b*x2 + c*x3 + k")
285     print("[a b c k] = ",end="")
286
287     indice = 0
288     while (indice < len(w1)):
289         coeficientes_com_aux.append(w1[indice][0])
290         coeficientes_com = np.array(coeficientes_com_aux)
291         indice += 1
292
293     print(coeficientes_com)
294     coeficientes_com_aux = []
295     indice = 0
296
297
298
299 if (resultado == '2'):
300     print("Digite qual especie voce deseja fazer a decomposicao
301           espectral: ")
302     print("1 = Iris-Setosa")
303     print("2 = Iris-Versicolor")
304     print("3 = Iris Virginica")
305     print()
306     print("PARA SAIR DIGITE 0")
307
308     while (tipo_iris != '0'):
309
310         tipo_iris = input()
311
312         if (tipo_iris == '0'):
313             break
314
315         print()
316
317         dados = pegarDados (tipo_iris)
318         R,p,R1,p1 = construir_equacao_normal(dados)
319         autovetores, matrizDiagonal = decomposicao_espectral(R)
320         autovetores1, matrizDiagonal1 = decomposicao_espectral(R1)
321
322         if (tipo_iris == '1'):
323             print("Iris-Setosa\n")

```

```

323         elif (tipo_iris == '2'):
324             print("Iris-Versicolor\n")
325         elif (tipo_iris == '3'):
326             print("Iris-Virginica\n")
327
328         print("R = V\u0392V^(T)")
329         print()
330
331         print("SEM O TERMO INDEPENDENTE: ")
332         print("V = ",autovetores)
333         print()
334         print('\u0392 = ',matrizDiagonal)
335         print()
336         print("V^T = ",np.transpose(autovetores))
337
338         print()
339         print()
340
341         print("COM O TERMO INDEPENDENTE: ")
342         print("V = ", autovetores1)
343         print()
344         print('\u0392 = ', matrizDiagonal1)
345         print()
346         print("V^T = ", np.transpose(autovetores1))
347
348
349
350     if (resultado == '3'):
351         print("Digite qual especie voce deseja fazer o SVD: ")
352         print("1 = Iris-Setosa")
353         print("2 = Iris-Versicolor")
354         print("3 = Iris Virginica")
355         print()
356         print("PARA SAIR DIGITE 0")
357
358         while (tipo_iris != '0'):
359             tipo_iris = input()
360
361             if (tipo_iris == '0'):
362                 break
363
364             print()
365
366             dados = pegarDados (tipo_iris)
367             R,p,R1,p1 = construir_equacao_normal(dados)
368             U, s, VT = linalg.svd(R)
369             U1, s1, VT1 = linalg.svd(R1)
370
371             if (tipo_iris == '1'):
372                 print("Iris-Setosa\n")
373             elif (tipo_iris == '2'):

```



```

374         print("Iris-Versicolor\n")
375     elif (tipo_iris == '3'):
376         print("Iris-Virginica\n")
377
378     print("R = U\u03A3V^(T)")
379     print()
380
381     print("SEM O TERMO INDEPENDENTE: ")
382     print("U = ", U)
383     print()
384     print('\u03A3 = ', diagsvd(s, R.shape[0], R.shape[1]))
385     print()
386     print("V^T = ", VT)
387     print()
388     print()
389
390     print("COM O TERMO INDEPENDENTE: ")
391     print("U = ", U1)
392     print()
393     print('\u03A3 = ', diagsvd(s1, R1.shape[0], R1.shape[1]))
394     print()
395     print("V^T = ", VT1)
396
397
398 if (resultado == '4'):
399     A = [5.0,2.3,3.3,1.0]
400     B = [4.6,3.2,1.4,0.2]
401     C = [5.0,3.3,1.4,0.2]
402     D = [6.1,3.0,4.6,1.4]
403     E = [5.9,3.0,5.1,1.8]
404
405     print("SEM O TERMO INDEPENDENTE: ")
406     dados = pegarDados ('1')
407     R,p,R1,p1 = construir_equacao_normal(dados)
408     w_setosa = PLU(R,p)
409
410     dados = pegarDados ('2')
411     R,p,R1,p1 = construir_equacao_normal(dados)
412     w_versicolor = PLU(R,p)
413
414     dados = pegarDados ('3')
415     R,p,R1,p1 = construir_equacao_normal(dados)
416     w_virginica = PLU(R,p)
417
418     c_independente = False
419     estimativa = estimar_amostras(A,w_setosa,w_versicolor,
420                                   w_virginica,c_independente)
421     print("A = ", estimativa)
422     estimativa = estimar_amostras(B,w_setosa,w_versicolor,
423                                   w_virginica,c_independente)
424     print("B = ", estimativa)

```

```

423     estimativa = estimar_amostras(C,w_setosa,w_versicolor,
424                                   w_virginica,c_independente)
425     print("C = ", estimativa)
426     estimativa = estimar_amostras(D,w_setosa,w_versicolor,
427                                   w_virginica,c_independente)
428     print("D = ", estimativa)
429     estimativa = estimar_amostras(E,w_setosa,w_versicolor,
430                                   w_virginica,c_independente)
431     print("E = ", estimativa)
432
433     print()
434     print("COM O TERMO INDEPENDENTE: ")
435     dados = pegarDados ('1')
436     R,p,R1,p1 = construir_equacao_normal(dados)
437     w1_setosa = PLU(R1,p1)
438
439     dados = pegarDados ('2')
440     R,p,R1,p1 = construir_equacao_normal(dados)
441     w1_versicolor = PLU(R1,p1)
442
443     dados = pegarDados ('3')
444     R,p,R1,p1 = construir_equacao_normal(dados)
445     w1_virginica = PLU(R1,p1)
446
447     c_independente = True
448     estimativa1 = estimar_amostras(A,w1_setosa,w1_versicolor,
449                                   w1_virginica,c_independente)
450     print("A = ", estimativa1)
451     estimativa1 = estimar_amostras(B,w1_setosa,w1_versicolor,
452                                   w1_virginica,c_independente)
453     print("B = ", estimativa1)
454     estimativa1 = estimar_amostras(C,w1_setosa,w1_versicolor,
455                                   w1_virginica,c_independente)
456     print("C = ", estimativa1)
457     estimativa1 = estimar_amostras(D,w1_setosa,w1_versicolor,
458                                   w1_virginica,c_independente)
459     print("D = ", estimativa1)
460     estimativa1 = estimar_amostras(E,w1_setosa,w1_versicolor,
461                                   w1_virginica,c_independente)
462     print("E = ", estimativa1)
463
464     ##### chamada ao menu
465     menu()

```

## 7 Bibliografia

- <https://algebrainearufcg.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>`scipy.linalg.diagsvd`