

## Trabalho - AV2

Fundamentos da programação 2021.2

### Grupo:

Gustavo Henrique  
(ghco@cesar.school)

Beatriz Vilarim  
(bsvs@cesar.school)

Júlia Boto  
(jbr@cesar.school)

### Resumo:

Desenvolvemos um código na linguagem Python capaz de interpretar o sentimento de uma pessoa com base em um texto e escrever num arquivo .csv assim como atualizar novos pesos de novas frases.

### 1. Frases:

Frases selecionadas e tabela de palavras com seus respectivos pesos e classes:

- 1.Eu estou feliz com a minha garrafa d'água. **(Boa)**
- 2.Eu estou feliz sobre como é fácil usar o meu produto e como ajudou o meu time a completar tarefas do dia a dia com sucesso. **(Boa)**
- 3.Totalmente insatisfeita com o serviço. Pior atendimento de todos. **(Ruim)**
- 4.A minha empresa tem um bom ambiente para se trabalhar, mas o salário é muito ruim. **(Neutra)**
- 5.Amei seu produto, irei comprar mais vezes. **(Boa)**
- 6.Perder o interesse em tudo porque

você está triste é o pior sentimento. **(Ruim)**

7.Uma pessoa que nunca cometeu um erro, nunca tentou nada de novo. **(Ruim)**

8.Hoje é um bom dia para ser feliz. **(Boa)**

9.Esqueça os pensamentos negativos e foque apenas no que é positivo. **(Boa)**

10.Produto de qualidade impecável, já é a minha segunda compra no site. **(Boa)**

### 1.1 Tabela:

Boas:	feliz (5), estou (2), minha (3), fácil (5), ajudou (5), sucesso (5), qualidade (5), esqueça (3), amei (5), todos (2)	5
Neutras:	Eu, com, a, um, que, tentou, ser, o, de, seu	0
Ruins:	Negativos (-5), erro (-3), nunca (-3), triste (-5), pior (-5), ruim (-3), usar (-2), interesse (-2), perder (-5), insatisfeita (-5)	-5

Classificamos as palavras sendo o peso 5 como boas, 0 como neutras e -5 como ruins.

### 2. Descrição do código:

```

38 palavras_reconhecidas = []
39 lista_boas = []
40 palavras_novas = []
41
42 # Boas
43
44 for i in frases[index_frase]:
45     if i in pesos[0]:
46         palavras_reconhecidas.append(i)
47         index = (pesos[0].index(i))+1
48         valor_boa = pesos[0][index]
49         lista_boas.append(valor_boa)
50
51 # Neutras
52
53 lista_neutras = []
54
55 for i in frases[index_frase]:
56     if i in pesos[2]:
57         palavras_reconhecidas.append(i)
58         index = (pesos[2].index(i))+1
59         valor_neutra = pesos[2][index]
60         lista_neutras.append(valor_neutra)
61
62 # Ruims
63
64 lista_ruims = []
65
66 for i in frases[index_frase]:
67     if i in pesos[4]:
68         palavras_reconhecidas.append(i)
69         index = (pesos[4].index(i))+1
70         valor_ruim = pesos[4][index]
71         lista_ruims.append(valor_ruim)
72

```

Após o tratamento do texto o programa inicia uma contagem das frases boas, neutras e ruins e armazena cada uma em uma lista com base na frase escolhida a ser processada (No caso a frase é dada pelo **index\_frase**)  
A classificação da palavra é baseada no arquivo de peso.

```

83 # Calculo de Análise de sentimento
84
85 qtd_palavras_rec = 0
86
87 for i in palavras_reconhecidas:
88     qtd_palavras_rec+=1
89
90 soma_boas = 0
91 soma_neutras = 0
92 soma_ruims = 0
93
94 for c in lista_boas:
95     soma_boas+=int(c)
96
97 for c in lista_neutras:
98     soma_neutras+=int(c)
99
100 for c in lista_ruims:
101     soma_ruims+=int(c)
102
103 media = round((soma_boas+soma_neutras+soma_ruims)/qtd_palavras_rec)
104
105 print(media)
106
107 classificacao = ''
108
109 if media < 0:
110     classificacao = 'Ruim'
111 if media == 0:
112     classificacao = 'Neutra'
113 if media > 0:
114     classificacao = 'Boa'
115

```

No código da análise de sentimento o programa conta a quantidade de palavras reconhecidas dentro da frase selecionada a ser processada e acumula em uma variável na sua classificação(boas, neutras ou ruins), Então é calculado a média para esta frase com base na sua quantidade de palavras reconhecidas, e depois a frase é classificada com base em condicionais a partir da linha 99 e com base no valor predominante de sua média. A variável **classificacao** recebe “**Boa, Neutra ou Ruim**”.

1)

```

73 # Palavras novas
74
75 pesos_tot = pesos[0]+pesos[2]+pesos[4]
76
77 for i in frases[index_frase]:
78     if i not in pesos_tot:
79         palavras_novas.append(i)
80
81 print('palavras novas {}'.format(palavras_novas))
82

```

2)

```

124 # Algoritmo para aprender novas palavras:
125
126 algoritmo = []
127
128 for i in palavras_novas:
129     algoritmo.append(i)
130     algoritmo.append(media)
131     algoritmo.append('')
132
133 res_algoritmo = ' '.join(str(e) for e in algoritmo)
134
135 with open("pesos.txt", "r") as file:
136     texto = file.readlines()
137
138 if classificacao == "Ruim":
139     texto.insert(4, res_algoritmo)
140 elif classificacao == "Boa":
141     texto.insert(0, res_algoritmo)
142 elif classificacao == "Neutra":
143     texto.insert(2, res_algoritmo)
144
145 with open("pesos.txt", "w") as file:
146     texto = "".join(texto)
147     file.write(texto)
148 print('> Novos pesos adicionados com sucesso! ')
149

```

Para separar as palavras novas, intercedemos (mesclamos) as palavras reconhecidas que ainda não estão dentro do arquivo de pesos, assim poderemos usar no código para aprender novas palavras, que a partir das palavras novas irá calcular a atribuir às novas frases no arquivo de pesos com base no seu sentimento predominante (boa, neutra ruim) junto com seu novo peso, com base na média calculada anteriormente.

## 2.2 Resultados

```

pesos.txt x
1 Boas: feliz 5 estou 2 minha 3 fácil 5 ajudou 5 sucesso 5 qualidade 5 esqueça 3 amei
2
3 Neutras: eu 0 com 0 a 0 um 0 que 0 tentou 0 ser 0 o 0 de 0 seu 0
4
5 Ruims: negativos -5 erro -3 nunca -3 triste -5 pior -5 ruim -3 usar -2 interesse -2
   insatisfeita -5

```

```

pesos.txt x
1 Boas: feliz 5 estou 2 minha 3 fácil 5 ajudou 5 sucesso 5 qualidade 5 esqueça 3 amei 5 todos 2
2
3 Neutras: eu 0 com 0 a 0 um 0 que 0 tentou 0 ser 0 o 0 de 0 seu 0
4
5 totalmente -1 serviço -1 atendimento -1 Ruims: negativos -5 erro -3 nunca -3 triste -5 pior -3
   ruim -3 usar -2 interesse -2 perder -5 insatisfeita -5

```

No print o programa consegue atualizar a lista de pesos com as novas palavras e seu novo peso, que poderá ser usados para as novas frases, validando o algoritmo.

```

resultados.csv x
1 Frase, Lista de palavras reconhecidas, Resultado
2
3 totalmente insatisfeita com o serviço pior a
Ruim, totalmente serviço atendimento,

```

E depois os novos resultados são adicionados no arquivo csv separados por vírgula.

	A	
1	Frase	Lista de pal
2		
3	totalmente insatisfeita com o serviço pior atendimento de todos	todos com c
4		

Na print a visualização do CSV na numa planilha, contendo todas as colunas com seus respectivos resultados.

### 3. Conclusão:

O grupo passou boa parte do tempo discutindo bastante como o cálculo poderia ser feito para ter a maior precisão possível assim como seria possível “Desviar” de possíveis dicotomias e oposições de conectivos nas frases ou até ironias, e chegamos a um resultado satisfatório nos cálculos de sentimento, que consegue definir bem o sentimento da frase em si.

O grupo achou interessante o usuário escolher a frase que ele quisesse que fosse processada aos poucos, pois assim seria possível ver com detalhes o algoritmo de aprendizado funcionando e adicionando as informações aos poucos nas saídas do programa, tanto no arquivo de pesos,

Quanto ao arquivo de csv, sem perigo de repetir frases, já que o algoritmo consegue separar de palavras novas, das que já existem.

Apesar de algumas limitações, como na atualização das palavras de pesos, onde o cursor não consegue escrever um “uma linha específica” do arquivo de texto, já que organizamos os arquivos de peso linha por linha com cada classificação (Boas, Neutras, Ruim) mas o programa funciona normalmente acessando os pesos no arquivo apesar de não visualmente agradável.

De fato foi uma experiência riquíssima e o grupo sentiu bastante aprendizado dentro do trabalho, além de ser muito interessante e divertido.

