NLP

February 27, 2021

1 Dados

```
(518, 2)
```

[2]:

	letra	artista
0	Jay-z Uh-uh-uh You ready b? Let's go get 'em. …	Beyoncé
1	Your challengers are a young group from Housto…	Beyoncé
2	Dum-da-de-da Do, do, do, do, do, do (Coming do…	Beyoncé
3	If I ain't got nothing I got you If I ain't go…	Beyoncé
4	Six inch heels She walked in the club like nob…	Beyoncé
5	(hello) hello How are you (oh) I just got to s…	Beyoncé
6	Shoulders sideways, smack it, smack it in the …	Beyoncé
7	Clap, clap, clap like you don't care Ooh we b…	Beyoncé
8	Shoulders sideways, smack it, smack it in the …	Beyoncé
9	Do you think You could fall for a woman like m…	Beyoncé

2 Pré-procesamento de texto

Preparando o dataset para a aplicação do algoritmo de naive Bayes

Exclusão de capitulares

[3]:
```
0    jay-z uh-uh-uh you ready b? let's go get 'em. …
1    your challengers are a young group from housto…
2    dum-da-de-da do, do, do, do, do, do (coming do…
3    if i ain't got nothing i got you if i ain't go…
4    six inch heels she walked in the club like nob…
5    (hello) hello how are you (oh) i just got to s…
6    shoulders sideways, smack it, smack it in the …
7    clap, clap, clap like you don't care  ooh we b…
8    shoulders sideways, smack it, smack it in the …
9    do you think you could fall for a woman like m…
Name: letra, dtype: object
```

Tokenização

[5]:
```
0    [jayz, uhuhuh, you, ready, b, lets, go, get, e…
1    [your, challengers, are, a, young, group, from…
2    [dumdadeda, do, do, do, do, do, do, coming, do…
3    [if, i, aint, got, nothing, i, got, you, if, i…
4    [six, inch, heels, she, walked, in, the, club,…
5    [hello, hello, how, are, you, oh, i, just, got…
6    [shoulders, sideways, smack, it, smack, it, in…
7    [clap, clap, clap, like, you, dont, care, ooh,…
8    [shoulders, sideways, smack, it, smack, it, in…
9    [do, you, think, you, could, fall, for, a, wom…
Name: letra, dtype: object
```

Remoção de 'stopwords'

[7]:
```
0    [jayz, uhuhuh, ready, b, lets, go, get, em, lo…
1    [challengers, young, group, houston, welcome, …
2    [dumdadeda, coming, dripping, candy, ground, s…
3    [aint, got, nothing, got, aint, got, something…
4    [six, inch, heels, walked, club, like, nobodys…
5    [hello, hello, oh, got, say, hold, know, thing…
6    [shoulders, sideways, smack, smack, air, legs,…
7    [clap, clap, clap, like, dont, care, ooh, frea…
8    [shoulders, sideways, smack, smack, air, legs,…
9    [think, could, fall, woman, like, cause, find,…
Name: letra, dtype: object
```

'Stemming' para a obtenção da raiz das palavras

[8]:
```
0    [jayz, uhuhuh, readi, b, let, go, get, em, loo…
1    [challeng, young, group, houston, welcom, beyo…
2    [dumdadeda, come, drip, candi, ground, stay, y…
3    [aint, got, noth, got, aint, got, someth, dont…
4    [six, inch, heel, walk, club, like, nobodi, bu…
5    [hello, hello, oh, got, say, hold, know, thing…
6    [shoulder, sideway, smack, smack, air, leg, mo…
7    [clap, clap, clap, like, dont, care, ooh, frea…
8    [shoulder, sideway, smack, smack, air, leg, mo…
9    [think, could, fall, woman, like, caus, find, …
Name: letra, dtype: object
```

Vetorização Gerando uma matriz TD-IDF

Entrada X

[11]:
```
     0         1     2         3     4     5     6     7     8     9     …  \
0   0.0  0.055606   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0   …
```

```
1     0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
2     0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
3     0.0  0.000000   0.0  0.14249   0.0   0.0   0.0   0.0   0.0   0.0  …
4     0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
..    …       …   …       …   …   …   …   …   …   …  …
513   0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
514   0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
515   0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
516   0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …
517   0.0  0.000000   0.0  0.00000   0.0   0.0   0.0   0.0   0.0   0.0  …

     6409  6410  6411  6412  6413  6414  6415  6416  6417  6418
0     0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
1     0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
2     0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
3     0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4     0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
..    …   …   …   …   …   …   …   …   …   …
513   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
514   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
515   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
516   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
517   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[518 rows x 6419 columns]
```

Saída y

[12]:

```
          0
0  Beyoncé
1  Beyoncé
2  Beyoncé
3  Beyoncé
4  Beyoncé
```

3 5. Crie um classificador, a partir da segunda aba - NLP do arquivo de dados, que permita identificar qual trecho de música corresponde às respectivas artistas listadas (Sugestão: Naive Bayes Classifier).

52.9% das letras são do rótulo mais frequente (Beyoncé), portanto, só fará sentido um classificador que acerte pelo menos 52.9% das previsões.

```
Beyoncé    0.528958
Rihanna    0.471042
```

```
Name: artista, dtype: float64
```

Split dos dados

3.0.1 Naive Bayes

```
Matriz de confusão: Observados (linhas) x Classificados (colunas)
```

[16]:

	Beyoncé	Rihanna
Beyoncé	39	6
Rihanna	46	65

```
Acurácia: 0.6667
```

3.0.2 SVM

```
Matriz de confusão: Observados (linhas) x Classificados (colunas)
```

[19]:

	Beyoncé	Rihanna
Beyoncé	74	37
Rihanna	11	34

```
Acurácia: 0.6923
```

3.0.3 Random forest

```
Matriz de confusão: Observados (linhas) x Classificados (colunas)
```

[22]:

	Beyoncé	Rihanna
Beyoncé	63	27
Rihanna	22	44

```
Acurácia: 0.6859
```

3.0.4 Redes neurais

```
Matriz de confusão: Observados (linhas) x Classificados (colunas)
```

[25]:

	Beyoncé	Rihanna
Beyoncé	59	23
Rihanna	26	48

```
Acurácia: 0.6859
```

3.1 Comparação

O algoritmo de melhor acurácia, com 69.29% de acertos, foi o SVM. Tentaremos melhorá-lo "tunando" os hiperparâmetros.

[27]:

	Acurácia
Naive bayes	0.6667
SVM	0.6923
Random forest	0.6859
Redes neurais	0.6859

3.2 Grid search para tuning dos parâmetros

```
Melhores parâmetros:
{'C': 3, 'gamma': 0.3, 'kernel': 'sigmoid'}

Acurácia:0.7244
```

3.3 Salvando o modelo

Após a melhora acima, podemos colocar em produção um modelo com 72.44% de acurácia.