

# TRABALHO DE IAA006 – Arquitetura de Dados

## 1 Construção de Características: Identificador automático de idioma

### Equipe 03

- Gustavo Costa de Souza
- Marcos Vinicius de Melo
- Marcus Eneas Silveira Galvao do Rio Apa II
- Patrícia Verdugo Pascoal
- Rodrigo de Araujo
- William de Souza Alencar

## Identificador automático de idioma

**Problema:** Dados um texto de entrada, é possível identificar em qual língua o texto está escrito?

Entrada: "texto qualquer"

Saída: português ou inglês ou francês ou italiano ou...

## O processo de Reconhecimento de Padrões

O objetivo desse trabalho é demonstrar o processo de "construção de atributos" e como ele é fundamental para o **Reconhecimento de Padrões (RP)**.

Primeiro um conjunto de "amostras" previamente conhecido (classificado)

```
In [1]: #  
# amostras de texto em diferentes línguas  
#  
ingles = [  
    "Hello, how are you?",  
    "I love to read books.",  
    "The weather is nice today.",  
    "Where is the nearest restaurant?",  
    "What time is it?",  
    "I enjoy playing soccer.",  
    "Can you help me with this?",  
    "I'm going to the movies tonight.",  
    "This is a beautiful place.",  
    "I like listening to music.",  
    "Do you speak English?",  
    "What is your favorite color?",  
    "I'm learning to play the guitar.",  
    "Have a great day!",  
    "I need to buy some groceries.",  
    "Let's go for a walk.",  
    "How was your weekend?",  
    "I'm excited for the concert.",  
    "Could you pass me the salt, please?",  
    "I have a meeting at 2 PM.",  
    "I'm planning a vacation.",  
    "She sings beautifully.",  
    "The cat is sleeping.",  
    "I want to learn French.",  
    "I enjoy going to the beach.",  
    "Where can I find a taxi?",  
    "I'm sorry for the inconvenience.",  
    "I'm studying for my exams.",  
    "I like to cook dinner at home.",  
    "Do you have any recommendations for restaurants?",  
]  
  
espanhol = [  
    "Hola, ¿cómo estás?",  
    "Me encanta leer libros.",  
    "El clima está agradable hoy.",
```

```
"¿Dónde está el restaurante más cercano?",  
"¿Qué hora es?",  
"Voy al parque todos los días.",  
"¿Puedes ayudarme con esto?",  
"Me gustaría ir de vacaciones.",  
"Este es mi libro favorito.",  
"Me gusta bailar salsa.",  
"¿Hablas español?",  
"¿Cuál es tu comida favorita?",  
"Estoy aprendiendo a tocar el piano.",  
"¡Que tengas un buen día!",  
"Necesito comprar algunas frutas.",  
"Vamos a dar un paseo.",  
"¿Cómo estuvo tu fin de semana?",  
"Estoy emocionado por el concierto.",  
"¿Me pasas la sal, por favor?",  
"Tengo una reunión a las 2 PM.",  
"Estoy planeando unas vacaciones.",  
"Ella canta hermosamente.",  
"El perro está jugando.",  
"Quiero aprender italiano.",  
"Disfruto ir a la playa.",  
"¿Dónde puedo encontrar un taxi?",  
"Lamento las molestias.",  
"Estoy estudiando para mis exámenes.",  
"Me gusta cocinar la cena en casa.",  
"¿Tienes alguna recomendación de restaurantes?",  
]  
  
portugueses = [  
"Estou indo para o trabalho agora.",  
"Adoro passar tempo com minha família.",  
"Preciso comprar leite e pão.",  
"Vamos ao cinema no sábado.",  
"Gosto de praticar esportes ao ar livre.",  
"O trânsito está terrível hoje.",  
"A comida estava deliciosa!",  
"Você já visitou o Rio de Janeiro?",  
"Tenho uma reunião importante amanhã.",  
"A festa começa às 20h.",  
"Estou cansado depois de um longo dia de trabalho.",
```

```
"Vamos fazer um churrasco no final de semana.",  
"O livro que estou lendo é muito interessante.",  
"Estou aprendendo a cozinhar pratos novos.",  
"Preciso fazer exercícios físicos regularmente.",  
"Vou viajar para o exterior nas férias.",  
"Você gosta de dançar?",  
"Hoje é meu aniversário!",  
"Gosto de ouvir música clássica.",  
"Estou estudando para o vestibular.",  
"Meu time de futebol favorito ganhou o jogo.",  
"Quero aprender a tocar violão.",  
"Vamos fazer uma viagem de carro.",  
"O parque fica cheio aos finais de semana.",  
"O filme que assisti ontem foi ótimo.",  
"Preciso resolver esse problema o mais rápido possível.",  
"Adoro explorar novos lugares.",  
"Vou visitar meus avós no domingo.",  
"Estou ansioso para as férias de verão.",  
"Gosto de fazer caminhadas na natureza.",  
"O restaurante tem uma vista incrível.",  
"Vamos sair para jantar no sábado.",  
]
```

A "amostras" de texto precisa ser "transformada" em **padrões**

Um padrão é um conjunto de características, geralmente representado por um vetor e um conjunto de padrões no formato de tabela. Onde cada linha é um padrão e as colunas as características e, geralmente, na última coluna a **classe**

```
In [2]: import random  
  
pre_padroes = []  
for frase in ingles:  
    pre_padroes.append( [frase, 'inglês'])  
  
for frase in espanhol:  
    pre_padroes.append( [frase, 'espanhol'])  
  
for frase in portugues:  
    pre_padroes.append( [frase, 'português'])
```

```
random.shuffle(pre_padroes)
print(pre_padroes)
```

```
[['Estou estudando para o vestibular.', 'português'], ['Disfruto ir a la playa.', 'espanhol'], ['Quero aprender a tocar violã
o.', 'português'], ['The cat is sleeping.', 'inglês'], ['O livro que estou lendo é muito interessante.', 'português'], ['What i
s your favorite color?', 'inglês'], ['Meu time de futebol favorito ganhou o jogo.', 'português'], ['Este es mi libro favorit
o.', 'espanhol'], ['I enjoy playing soccer.', 'inglês'], ['Vamos a dar un paseo.', 'espanhol'], ['El perro está jugando.', 'esp
anhol'], ["I'm excited for the concert.", 'inglês'], ['Adoro passar tempo com minha família.', 'português'], ['Estou indo para
o trabalho agora.', 'português'], ['El clima está agradable hoy.', 'espanhol'], ['Vou viajar para o exterior nas férias.', 'por
tuguês'], ['Where is the nearest restaurant?', 'inglês'], ['Preciso comprar leite e pão.', 'português'], ['Could you pass me th
e salt, please?', 'inglês'], ['¿Hablas español?', 'espanhol'], ['¿Cómo estuvo tu fin de semana?', 'espanhol'], ['Estou ansioso
para as férias de verão.', 'português'], ['Where can I find a taxi?', 'inglês'], ['I like listening to music.', 'inglês'], ['Es
toy planeando unas vacaciones.', 'espanhol'], ['Tenho uma reunião importante amanhã.', 'português'], ['Me encanta leer libro
s.', 'espanhol'], ['O parque fica cheio aos finais de semana.', 'português'], ['Tengo una reunión a las 2 PM.', 'espanhol'],
['Voy al parque todos los días.', 'espanhol'], ['Me gusta bailar salsa.', 'espanhol'], ['Vamos fazer um churrasco no final de s
emana.', 'português'], ['Hello, how are you?', 'inglês'], ['Lamento las molestias.', 'espanhol'], ['¿Tienes alguna recomendaci
ón de restaurantes?', 'espanhol'], ["Let's go for a walk.", 'inglês'], ['I enjoy going to the beach.', 'inglês'], ["I'm studying
for my exams.", 'inglês'], ['Hola, ¿cómo estás?', 'espanhol'], ['This is a beautiful place.', 'inglês'], ['¿Que tengas un buen
día!', 'espanhol'], ['I need to buy some groceries.', 'inglês'], ['She sings beautifully.', 'inglês'], ['How was your weeken
d?', 'inglês'], ['Vamos sair para jantar no sábado.', 'português'], ['Adoro explorar novos lugares.', 'português'], ['Quiero ap
render italiano.', 'espanhol'], ['The weather is nice today.', 'inglês'], ['Do you speak English?', 'inglês'], ['¿Me pasas la s
al, por favor?', 'espanhol'], ['I love to read books.', 'inglês'], ['Hoje é meu aniversário!', 'português'], ['O restaurante te
m uma vista incrível.', 'português'], ['Vamos fazer uma viagem de carro.', 'português'], ['O filme que assisti ontem foi ótim
o.', 'português'], ['¿Dónde puedo encontrar un taxi?', 'espanhol'], ['I have a meeting at 2 PM.', 'inglês'], ['Me gusta cocinar
la cena en casa.', 'espanhol'], ['I want to learn French.', 'inglês'], ['¿Puedes ayudarme con esto?', 'espanhol'], ['Me gustarí
a ir de vacaciones.', 'espanhol'], ['¿Cuál es tu comida favorita?', 'espanhol'], ["I'm sorry for the inconvenience.", 'inglê
s'], ['Preciso resolver esse problema o mais rápido possível.', 'português'], ['Do you have any recommendations for restaurant
s?', 'inglês'], ['Você gosta de dançar?', 'português'], ["I'm learning to play the guitar.", 'inglês'], ['Gosto de praticar esp
ortes ao ar livre.', 'português'], ['A comida estava deliciosa!', 'português'], ['Estoy aprendiendo a tocar el piano.', 'espanh
ol'], ['A festa começa às 20h.', 'português'], ['Estou cansado depois de um longo dia de trabalho.', 'português'], ["I'm going
to the movies tonight.", 'inglês'], ['Vamos ao cinema no sábado.', 'português'], ['Gosto de ouvir música clássica.', 'portuguê
s'], ['Necesito comprar algunas frutas.', 'espanhol'], ['Estou aprendendo a cozinhar pratos novos.', 'português'], ["I'm planni
ng a vacation.", 'inglês'], ['Estoy estudiando para mis exámenes.', 'espanhol'], ['Vou visitar meus avós no domingo.', 'portugu
ês'], ['O trânsito está terrível hoje.', 'português'], ['Have a great day!', 'inglês'], ['¿Dónde está el restaurante más cercan
o?', 'espanhol'], ['¿Qué hora es?', 'espanhol'], ['Can you help me with this?', 'inglês'], ['Ella canta hermosamente.', 'espanh
ol'], ['What time is it?', 'inglês'], ['Você já visitou o Rio de Janeiro?', 'português'], ['Preciso fazer exercícios físicos re
gularmente.', 'português'], ['Estoy emocionado por el concierto.', 'espanhol'], ['I like to cook dinner at home.', 'inglês'],
['Gosto de fazer caminhadas na natureza.', 'português']]
```

O DataFrame do pandas facilita a visualização.

```
In [3]: import pandas as pd
dados = pd.DataFrame(pre_padroes)
dados
```

```
Out[3]:
```

	0	1
0	Estou estudando para o vestibular.	português
1	Disfruto ir a la playa.	espanhol
2	Quero aprender a tocar violão.	português
3	The cat is sleeping.	inglês
4	O livro que estou lendo é muito interessante.	português
...	...	...
87	Você já visitou o Rio de Janeiro?	português
88	Preciso fazer exercícios físicos regularmente.	português
89	Estoy emocionado por el concierto.	espanhol
90	I like to cook dinner at home.	inglês
91	Gosto de fazer caminhadas na natureza.	português

92 rows × 2 columns

```
In [4]: # A criação de ngrams com o modelo TfidfVectorizer foi uma tentativa, porém não melhorou a acurácia do treinamento, por isso n
import numpy as np
import re
from sklearn.feature_extraction.text import TfidfVectorizer

pre_padroes_np = np.array(pre_padroes)
pattern_regex = re.compile('[^\w+]', re.UNICODE)
corpus = [re.sub(pattern_regex, ' ', str(item)) for item in pre_padroes_np[:, 0]]
print(corpus)
```

```

ngram_model = TfidfVectorizer(analyzer='char', ngram_range=(3, 5))
ngram_model.fit_transform(corpus)

```

```

['Estou estudando para o vestibular ', 'Disfruto ir a la playa ', 'Quero aprender a tocar violão ', 'The cat is sleeping ', 'O
livro que estou lendo é muito interessante ', 'What is your favorite color ', 'Meu time de futebol favorito ganhou o jogo ', 'E
ste es mi libro favorito ', 'I enjoy playing soccer ', 'Vamos a dar un paseo ', 'El perro está jugando ', 'I m excited for the
concert ', 'Adoro passar tempo com minha família ', 'Estou indo para o trabalho agora ', 'El clima está agradable hoy ', 'Vou v
iajar para o exterior nas férias ', 'Where is the nearest restaurant ', 'Preciso comprar leite e pão ', 'Could you pass me the
salt please ', ' Hablas español ', ' Cómo estuvo tu fin de semana ', 'Estou ansioso para as férias de verão ', 'Where can I fi
nd a taxi ', 'I like listening to music ', 'Estoy planeando unas vacaciones ', 'Tenho uma reunião importante amanhã ', 'Me enca
nta leer libros ', 'O parque fica cheio aos finais de semana ', 'Tengo una reunión a las 2 PM ', 'Voy al parque todos los días
', 'Me gusta bailar salsa ', 'Vamos fazer um churrasco no final de semana ', 'Hello how are you ', 'Lamento las molestias ', '
Tienes alguna recomendación de restaurantes ', 'Let s go for a walk ', 'I enjoy going to the beach ', 'I m studying for my exam
s ', 'Hola cómo estás ', 'This is a beautiful place ', ' Que tengas un buen día ', 'I need to buy some groceries ', 'She sing
s beautifully ', 'How was your weekend ', 'Vamos sair para jantar no sábado ', 'Adoro explorar novos lugares ', 'Quiero aprende
r italiano ', 'The weather is nice today ', 'Do you speak English ', ' Me pasas la sal por favor ', 'I love to read books ',
'Hoje é meu aniversário ', 'O restaurante tem uma vista incrível ', 'Vamos fazer uma viagem de carro ', 'O filme que assisti on
tem foi ótimo ', ' Dónde puedo encontrar un taxi ', 'I have a meeting at 2 PM ', 'Me gusta cocinar la cena en casa ', 'I want t
o learn French ', ' Puedes ayudarme con esto ', 'Me gustaría ir de vacaciones ', ' Cuál es tu comida favorita ', 'I m sorry for
the inconvenience ', 'Preciso resolver esse problema o mais rápido possível ', 'Do you have any recommendations for restaurants
', 'Você gosta de dançar ', 'I m learning to play the guitar ', 'Gosto de praticar esportes ao ar livre ', 'A comida estava del
iciosa ', 'Estoy aprendiendo a tocar el piano ', 'A festa começa às 20h ', 'Estou cansado depois de um longo dia de trabalho ',
'I m going to the movies tonight ', 'Vamos ao cinema no sábado ', 'Gosto de ouvir música clássica ', 'Necesito comprar algunas
frutas ', 'Estou aprendendo a cozinhar pratos novos ', 'I m planning a vacation ', 'Estoy estudiando para mis exámenes ', 'Vou
visitar meus avós no domingo ', 'O trânsito está terrível hoje ', 'Have a great day ', ' Dónde está el restaurante más cercano
', ' Qué hora es ', 'Can you help me with this ', 'Ella canta hermosamente ', 'What time is it ', 'Você já visitou o Rio de Jan
eiro ', 'Preciso fazer exercícios físicos regularmente ', 'Estoy emocionado por el concierto ', 'I like to cook dinner at home
', 'Gosto de fazer caminhadas na natureza ']

```

```

Out[4]: <Compressed Sparse Row sparse matrix of dtype 'float64'
        with 7296 stored elements and shape (92, 4748)>

```

## Construção dos atributos

Esse é o coração desse trabalho e que deverá ser desenvolvido por vocês. Pensem em como podemos "medir" cada frase/sentença e extrair características que melhorem o resultado do processo de identificação.

Após a criação de cada novo atributo, execute as etapas seguintes e registre as métricas da matriz de confusão. Principalmente acurácia e a precisão.

```
In [5]: # a entrada é o vetor pre_padroes e a saída desse passo deverá ser "padrões"
import re

from scipy.sparse import hstack

def tamanhoMedioFrases(texto):
    palavras = re.split("\s", texto)
    #print(palavras)
    tamanhos = [len(s) for s in palavras if len(s)>0]
    #print(tamanhos)
    soma = 0
    for t in tamanhos:
        soma=soma+t
    return soma / len(tamanhos)

def caracteresEspeciais(frase):
    return 1 if any(ord(char) > 127 for char in frase) else 0

def possuiDoisSimbolosSucessivos(frase):
    frase = frase.lower();
    for i in range(len(frase) - 1):
        if frase[i] == frase[i+1]:
            return 1;
    return 0;

def proporcaoDeSufixos(frase):
    sufixos_por_idioma = {
        0: ["mente", "dade", "eiro", "ista", "oso", "ável", "ível", "ência", "idade", "ização", "amento", "imento", "ar", "adc",
        1: ["ción", "ciones", "dad", "tad", "ero", "encia", "iendo", "amiento", "imientto", "mente", "oy", "ando", "nte", "nta"
        2: ["ing", "ed", "ly", "ness", "ment", "able", "ible", "tion", "sion", "ous"] # Inglês
    }
    palavras = frase.lower().split()
    contador = [0.0,0.0,0.0]
    for palavra in palavras:
        for posicao, sufixos in sufixos_por_idioma.items():
```



```

        for sufixo in sufixos:
            if palavra.endswith(sufixo):
                contador[posicao] += 1
total_palavras = len(palavras)
for i in range(len(contador)):
    if (contador[i] != 0):
        contador[i] /= total_palavras
return contador

# retorna um array com a contagem de cada lingua
# posição 0 - pt, posição 1 - es, posição 2 - en
# Logo a função será quebrada em 3 características
def contagemPalavrasFrequentes(frase):
    palavras_pt = ["que", "não", "porém", "então", "porque", "lá", "saúde", "para", "de", "no", "ao", "estou", "você", "um",
    palavras_es = ["no", "pero", "entonces", "el", "la", "es", "un", "estoy", "cómo", "tu", "gusta", "cuál", "ella", "de", "en
    palavras_en = ["the", "and", "but", "because", "there", "add", "sadness", "you", "i m", "to", "this", "have", "of", "in",

    frase = frase.lower();
    palavras = frase.lower().split()
    contador = [0,0,0]
    for p in palavras:
        if p in palavras_pt:
            contador[0] += 1
        if p in palavras_es:
            contador[1] += 1
        if p in palavras_en:
            contador[2] += 1
    return contador

def proporcaoAcentosPtEs(frase):
    acentos_pt_es = ['â', 'ê', 'ô', 'ã', 'õ', 'á', 'é', 'í', 'ó', 'ú', 'à']
    frase = frase.lower();
    total_acentos = sum(frase.count(c) for c in acentos_pt_es)
    return total_acentos / len(frase)

def criaNGrans(frase):
    x = ngram_model.transform([frase])
    return x.toarray()

def extraiCaracteristicas(frase):
    # frase é um vetor [ 'texto', 'lingua' ]

```

```

texto = frase[0]
#print(texto)
pattern_regex = re.compile('[^\w+]', re.UNICODE)
texto = re.sub(pattern_regex, ' ', texto)
#print(texto)
caracteristica1=tamanhoMedioFrases(texto)
caracteristica2=caracteresEspeciais(texto)
caracteristica3=possuiDoisSimbolosSucessivos(texto)
caracteristica4=proporcaoAcentosPtEs(texto)
caracteristica6, caracteristica7, caracteristica8 = contagemPalavrasFrequentes(texto)
caracteristica9, caracteristica10, caracteristica11 = proporcaoDeSufixos(texto)
# crescente as suas funcoes no vetor padrao
padrao = [caracteristica1, caracteristica2, caracteristica3, caracteristica4, caracteristica6, caracteristica7, caracteristica8, caracteristica9, caracteristica10, caracteristica11]
padrao_array = np.array(padrao)
return padrao

def geraPadroes(frases):
    padroes = []
    for frase in frases:
        padrao = extraiCaracteristicas(frase)
        padroes.append(padrao)
    return padroes

# converte o formato [frase classe] em
# [caracteristica_1, caracteristica_2,... caracteristica n, classe]
padroes = geraPadroes(pre_padroes)

#
# apenas para visualizacao
print(padroes)

dados = pd.DataFrame(padroes)
dados

```

[[5.8, 0, 0, 0.0, 3, 0, 0, 0.2, 0.2, 0.0, 'português'], [3.6, 0, 0, 0.0, 1, 1, 1, 0.0, 0.0, 0.0, 'espanhol'], [5.0, 1, 0, 0.0333333333333333, 1, 0, 1, 0.2, 0.2, 0.0, 'português'], [4.0, 0, 1, 0.0, 0, 0, 1, 0.0, 0.0, 0.25, 'inglês'], [4.625, 1, 1, 0.0222222222222223, 3, 1, 0, 0.0, 0.125, 0.0, 'português'], [4.6, 0, 0, 0.0, 0, 0, 0, 0.0, 0.0, 0.0, 'inglês'], [4.375, 0, 0, 0.0, 2, 1, 0, 0.0, 0.0, 0.0, 'português'], [4.2, 0, 0, 0.0, 0, 1, 0, 0.0, 0.0, 0.0, 'espanhol'], [4.75, 0, 1, 0.0, 0, 0, 0, 0.0, 0.25, 'inglês'], [3.2, 0, 0, 0.0, 1, 1, 1, 0.2, 0.0, 0.0, 'espanhol'], [4.5, 1, 1, 0.045454545454545456, 0, 1, 0, 0.0, 0.25, 0.0, 'espanhol'], [3.6666666666666665, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.16666666666666666, 'inglês'], [5.166666666666667, 1, 1, 0.02702702702702703, 0, 0, 0, 0.16666666666666666, 0.0, 0.0, 'português'], [4.5, 0, 0, 0.0, 3, 0, 0, 0.0, 0.0, 0.0, 'português'], [4.6, 1, 0, 0.03571428571428571, 0, 1, 0, 0.0, 0.2, 0.2, 'espanhol'], [4.428571428571429, 1, 0, 0.02631578947368421, 2, 0, 0, 0.14285714285714285, 0.0, 0.0, 'português'], [5.4, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [4.6, 1, 0, 0.03571428571428571, 1, 0, 0, 0.2, 0.0, 0.0, 'português'], [3.857142857142857, 0, 1, 0.0, 0, 0, 2, 0.0, 0.0, 0.0, 'inglês'], [6.5, 1, 0, 0.0, 0, 0, 0, 0.0, 0.0, 0.0, 'espanhol'], [3.8333333333333335, 1, 0, 0.03333333333333333, 1, 3, 0, 0.0, 0.0, 0.0, 'espanhol'], [4.428571428571429, 1, 0, 0.05263157894736842, 4, 1, 0, 0.14285714285714285, 0.0, 0.0, 'português'], [3.0, 0, 0, 0.0, 1, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [4.2, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.2, 'inglês'], [7.0, 0, 0, 0.0, 0, 1, 0, 0.0, 0.75, 0.0, 'espanhol'], [6.2, 1, 0, 0.055555555555555555, 1, 0, 0, 0.0, 0.2, 0.0, 'português'], [4.75, 0, 1, 0.0, 0, 0, 0, 0.0, 0.25, 0.0, 'espanhol'], [4.125, 0, 0, 0.0, 2, 1, 0, 0.0, 0.0, 0.0, 'português'], [3.142857142857143, 1, 0, 0.034482758620689655, 1, 1, 1, 0.0, 0.0, 0.0, 'espanhol'], [3.8333333333333335, 1, 0, 0.034482758620689655, 0, 1, 0, 0.0, 0.16666666666666666, 0.0, 'espanhol'], [4.5, 0, 0, 0.0, 0, 1, 0, 0.25, 0.0, 0.0, 'espanhol'], [4.5, 0, 1, 0.0, 3, 2, 0, 0.0, 0.0, 0.0, 'português'], [3.5, 0, 1, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [6.333333333333333, 0, 0, 0.0, 0, 1, 0, 0.3333333333333333, 0.0, 0.0, 'espanhol'], [7.8, 1, 0, 0.0222222222222223, 1, 1, 0, 0.0, 0.2, 0.0, 'espanhol'], [2.3333333333333335, 0, 0, 0.0, 1, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [3.5, 0, 0, 0.0, 0, 0, 2, 0.0, 0.16666666666666666, 0.16666666666666666, 'inglês'], [3.3333333333333335, 0, 0, 0.0, 0, 0, 0, 0.0, 0.0, 0.16666666666666666, 'inglês'], [4.333333333333333, 1, 1, 0.1111111111111111, 0, 1, 0, 0.0, 0.0, 0.0, 'espanhol'], [4.2, 0, 0, 0.0, 1, 0, 2, 0.0, 0.0, 0.0, 'inglês'], [3.6, 1, 0, 0.04166666666666664, 1, 2, 0, 0.0, 0.0, 0.0, 'espanhol'], [3.8333333333333335, 0, 1, 0.0, 0, 0, 1, 0.0, 0.0, 0.16666666666666666, 'inglês'], [6.333333333333333, 0, 1, 0.0, 0, 0, 0, 0.0, 0.0, 0.3333333333333333, 'inglês'], [4.25, 0, 1, 0.0, 0, 0, 0, 0.0, 0.0, 0.0, 'inglês'], [4.5, 1, 0, 0.030303030303030304, 2, 1, 0, 0.3333333333333333, 0.0, 0.0, 'português'], [6.25, 0, 0, 0.0, 0, 0, 0, 0.25, 0.0, 0.0, 'português'], [7.333333333333333, 0, 0, 0.0, 0, 0, 0, 0.0, 0.3333333333333333, 0.0, 'espanhol'], [4.2, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [4.25, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [3.3333333333333335, 0, 1, 0.0, 0, 1, 0, 0.0, 0.0, 0.0, 'espanhol'], [3.2, 0, 1, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [4.75, 1, 0, 0.08695652173913043, 0, 0, 0, 0.0, 0.0, 0.0, 'português'], [5.166666666666667, 1, 0, 0.02702702702702703, 2, 0, 0, 0.3333333333333333, 0.16666666666666666, 0.0, 'português'], [4.333333333333333, 0, 1, 0.0, 2, 1, 0, 0.0, 0.0, 0.0, 'português'], [4.142857142857143, 1, 1, 0.027777777777777776, 2, 1, 0, 0.0, 0.0, 0.0, 'português'], [5.0, 1, 0, 0.03225806451612903, 0, 1, 0, 0.2, 0.0, 0.0, 'espanhol'], [2.5714285714285716, 0, 1, 0.0, 1, 0, 2, 0.0, 0.0, 0.14285714285714285, 'inglês'], [3.7142857142857144, 0, 0, 0.0, 0, 3, 0, 0.14285714285714285, 0.0, 0.0, 'espanhol'], [3.6, 0, 0, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [5.25, 0, 0, 0.0, 0, 0, 0, 0.0, 0.0, 0.0, 'espanhol'], [4.8, 1, 0, 0.034482758620689655, 1, 1, 0, 0.0, 0.2, 0.0, 'espanhol'], [4.4, 1, 0, 0.03571428571428571, 0, 3, 0, 0.0, 0.0, 0.0, 'espanhol'], [4.333333333333333, 0, 1, 0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [5.75, 1, 1, 0.037037037037037035, 1, 0, 0, 0.125, 0.0, 0.0, 'português'], [5.857142857142857, 0, 1, 0.0, 0, 0, 2, 0.0, 0.0, 0.0, 'inglês'], [4.25, 1, 0, 0.047619047619047616, 2, 1, 0, 0.25, 0.0, 0.0, 'português'], [3.5714285714285716, 0, 0, 0.0, 0, 0, 2, 0.14285714285714285, 0.0, 0.14285714285714285, 'inglês'], [4.571428571428571, 0, 0, 0.0, 2, 1, 0, 0.2857142857142857, 0.0, 0.0, 'português'], [5.5, 0, 0, 0.0, 1, 0, 1, 0.0, 0.0, 0.0, 'português'], [4.833333333333333, 0, 0, 0.0, 1, 2, 1, 0.16666666666666666, 0.3333333333333333, 0.0, 'espanhol'], [3.4, 1, 0, 0.045454545454545456, 1, 0, 1, 0.0, 0.0, 0.0, 'português'], [4.4444444444444445, 0, 0, 0.0, 4, 2, 0, 0.1111111111111111, 0.0, 0.0, 'português'], [3.5714285714285716, 0, 0, 0.0, 0, 0, 2, 0.0, 0.0, 0.14285714285714285, 'inglês'], [4.2, 1, 0, 0.038461538461538464, 2, 1, 0, 0.2, 0.0, 0.0, 'português'],

```
[5.2, 1, 1, 0.06451612903225806, 1, 1, 0, 0.0, 0.0, 0.0, 'português'], [7.0, 0, 0, 0.0, 0, 0, 0, 0.25, 0.0, 0.0, 'espanhol'],
[5.833333333333333, 0, 0, 0.0, 2, 0, 1, 0.16666666666666666, 0.0, 0.0, 'português'], [3.8, 0, 1, 0.0, 1, 0, 1, 0.0, 0.0, 0.4,
'inglês'], [6.0, 1, 0, 0.02857142857142857, 1, 1, 0, 0.0, 0.4, 0.0, 'espanhol'], [4.5, 1, 0, 0.030303030303030304, 1, 1, 0, 0.1
6666666666666666, 0.0, 0.0, 'português'], [5.0, 1, 1, 0.1, 1, 0, 0, 0.2, 0.0, 0.0, 'português'], [3.25, 0, 0, 0.0, 1, 0, 2, 0.
0, 0.0, 0.0, 'inglês'], [5.333333333333333, 1, 0, 0.07692307692307693, 0, 1, 0, 0.0, 0.16666666666666666, 0.0, 'espanhol'], [3.
0, 1, 0, 0.07692307692307693, 0, 1, 0, 0.0, 0.0, 0.0, 'espanhol'], [3.3333333333333335, 0, 0, 0.0, 0, 0, 2, 0.0, 0.0, 0.0, 'ing
lês'], [7.0, 0, 1, 0.0, 0, 1, 0, 0.3333333333333333, 1.0, 0.0, 'espanhol'], [3.0, 0, 0, 0.0, 0, 0, 0, 0.0, 0.0, 0.0, 'inglês'],
[3.7142857142857144, 1, 0, 0.06060606060606061, 3, 1, 0, 0.14285714285714285, 0.0, 0.0, 'português'], [8.2, 1, 0, 0.04347826086
9565216, 0, 0, 0, 0.2, 0.4, 0.0, 'português'], [5.8, 0, 0, 0.0, 0, 2, 0, 0.2, 0.2, 0.0, 'espanhol'], [3.2857142857142856, 0, 1,
0.0, 0, 0, 1, 0.0, 0.0, 0.0, 'inglês'], [5.333333333333333, 0, 0, 0.0, 1, 1, 0, 0.0, 0.0, 0.0, 'português']]
```

```
Out[5]:
```

	0	1	2	3	4	5	6	7	8	9	10
0	5.800000	0	0	0.000000	3	0	0	0.200000	0.200	0.00	português
1	3.600000	0	0	0.000000	1	1	1	0.000000	0.000	0.00	espanhol
2	5.000000	1	0	0.033333	1	0	1	0.200000	0.200	0.00	português
3	4.000000	0	1	0.000000	0	0	1	0.000000	0.000	0.25	inglês
4	4.625000	1	1	0.022222	3	1	0	0.000000	0.125	0.00	português
...	...	...	...	...	...	...	...	...	...	...	...
87	3.714286	1	0	0.060606	3	1	0	0.142857	0.000	0.00	português
88	8.200000	1	0	0.043478	0	0	0	0.200000	0.400	0.00	português
89	5.800000	0	0	0.000000	0	2	0	0.200000	0.200	0.00	espanhol
90	3.285714	0	1	0.000000	0	0	1	0.000000	0.000	0.00	inglês
91	5.333333	0	0	0.000000	1	1	0	0.000000	0.000	0.00	português

92 rows × 11 columns

## Treinando o modelo com SVM

### Separando o conjunto de treinamento do conjunto de testes

```
In [6]: from sklearn.model_selection import train_test_split
import numpy as np

#from sklearn.metrics import confusion_matrix

vet = np.array(padroes)
classes = vet[:, -1]      # classes = [p[-1] for p in padroes]
#print(classes)
padroes_sem_classe = vet[:, 0:-1]
#print(padroes_sem_classe)
X_train, X_test, y_train, y_test = train_test_split(padroes_sem_classe, classes, test_size=0.25, stratify=classes, random_stat
```

Com os conjuntos separados, podemos "treinar" o modelo usando a SVM.

```
In [7]: from sklearn import svm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

treinador = svm.SVC(random_state=42) #algoritmo escolhido
modelo = treinador.fit(X_train, y_train)

#
# score com os dados de treinamento
acuracia = modelo.score(X_train, y_train)
print("Acurácia nos dados de treinamento: {:.2f}%".format(acuracia * 100))

#
# melhor avaliar com a matriz de confusão
y_pred = modelo.predict(X_train)
cm = confusion_matrix(y_train, y_pred)
print(cm)
print(classification_report(y_train, y_pred))

#
# com dados de teste que não foram usados no treinamento
print('métricas mais confiáveis')
y_pred2 = modelo.predict(X_test)
cm = confusion_matrix(y_test, y_pred2)
```

```
print(cm)
print(classification_report(y_test, y_pred2))
```

Acurácia nos dados de treinamento: 91.30%

```
[[20  1  1]
 [ 0 23  0]
 [ 3  1 20]]
```

	precision	recall	f1-score	support
espanhol	0.87	0.91	0.89	22
inglês	0.92	1.00	0.96	23
português	0.95	0.83	0.89	24
accuracy			0.91	69
macro avg	0.91	0.91	0.91	69
weighted avg	0.92	0.91	0.91	69

métricas mais confiáveis

```
[[8 0 0]
 [1 6 0]
 [2 0 6]]
```

	precision	recall	f1-score	support
espanhol	0.73	1.00	0.84	8
inglês	1.00	0.86	0.92	7
português	1.00	0.75	0.86	8
accuracy			0.87	23
macro avg	0.91	0.87	0.87	23
weighted avg	0.91	0.87	0.87	23