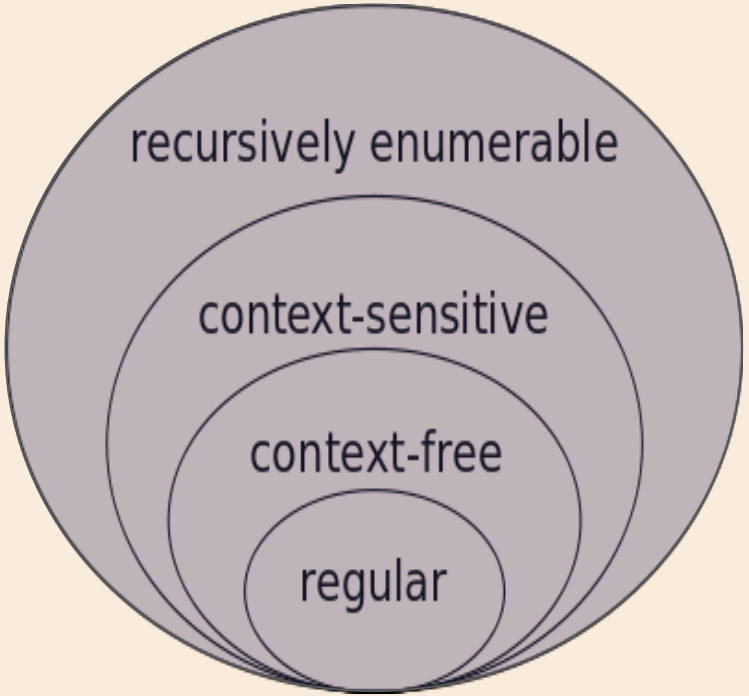
EXPRESSÕES REGULARES

PARA CIÊNCIA DE DADOS

O que é Expressão Regular?



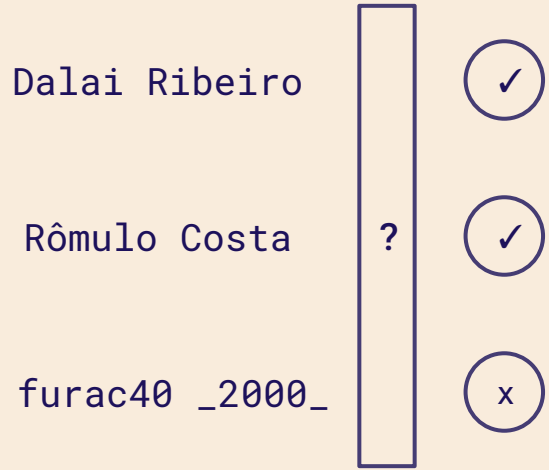
https://medium.com/@chetcorcos/introduction-to-parsers-644d1b5d7f3d

O que é Expressão Regular?

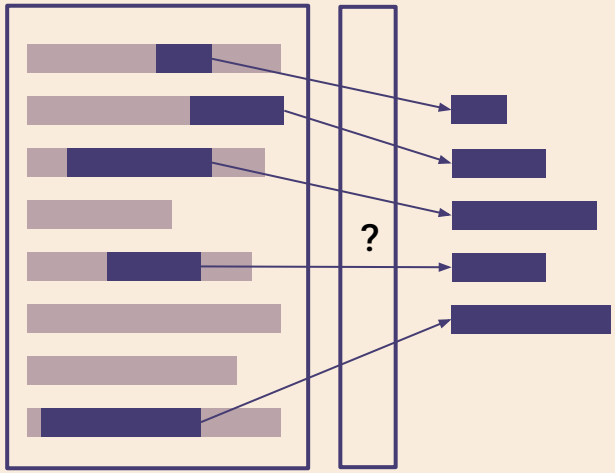
Caracteres comuns e especiais

Por que usar expressão regular?

Validação de dados

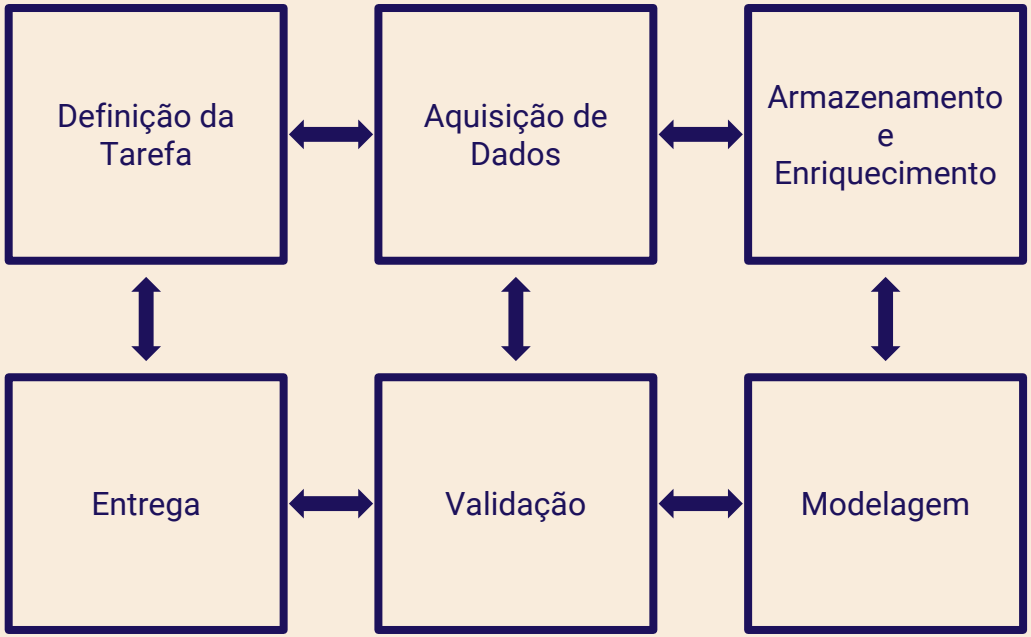


Extração de dados

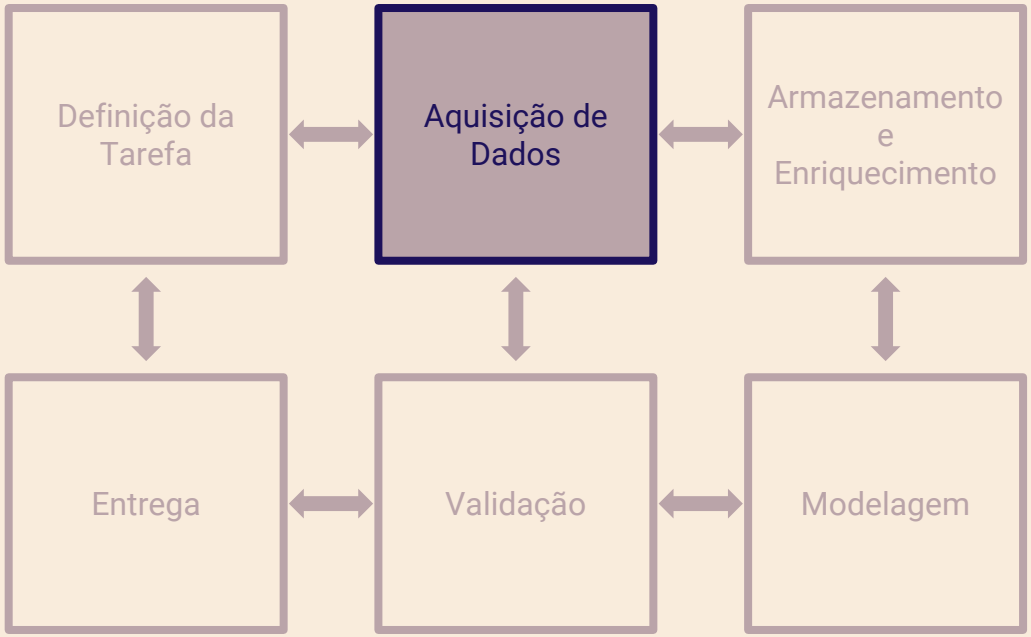


Expressão regular na Ciência de Dados

Ciclo da Ciência de Dados



Ciclo da Ciência de Dados



pré-processamento de texto

Principais Funções

Básico

```
a     # a
8     # 8
ab    # a diretamente
        seguido por b
```

Quantificadores

```
abc?     # ab seguido por 0..1 c
abc*     # ab seguido por 0..∞ c
abc+     # ab seguido por 1..∞ c
abc{3}   # ab seguido por 3 c
```

Quantificadores exemplos

	abc	abbc	ac	aaabc	bc	ab	abccc
ab?c							
a*bc							
abc+							
a{3}bc							

Quantificadores exemplos

	abc	abbc	ac	aaabc	bc	ab	abccc
ab?c	sim	não	sim	não	não	não	não
a*bc							
abc+							
a{3}bc							

Quantificadores exemplos

	abc	abbc	ac	aaabc	bc	ab	abccc
ab?c	sim	não	sim	não	não	não	não
a*bc	sim	não	não	sim	sim	não	não
abc+							
a{3}bc							

Quantificadores exemplos

	abc	abbc	ac	aaabc	bc	ab	abccc
ab?c	sim	não	sim	não	não	não	não
a*bc	sim	não	não	sim	sim	não	não
abc+	sim	não	não	não	não	não	sim
a{3}bc							

Quantificadores exemplos

	abc	abbc	ac	aaabc	bc	ab	abccc
ab?c	sim	não	sim	não	não	não	não
a*bc	sim	não	não	sim	sim	não	não
abc+	sim	não	não	não	não	não	sim
a{3}bc	não	não	não	sim	não	não	não

Agrupamento

```
(abc)+    # 1..∞ abc
(a|b)c    # ac OU bc
```

Agrupamento - exemplo

	abcde	de	be	c	abc	ababc
(abc)?de						
(a|b|c|d)e						
(a|b)*c						
(a|b)+c						

Agrupamento - exemplo

	abcde	de	be	c	abc	ababc
(abc)?de	sim	sim	não	não	não	não
(a|b|c|d)e						
(a|b)*c						
(a|b)+c						

Agrupamento - exemplo

	abcde	de	be	c	abc	ababc
(abc)?de	sim	sim	não	não	não	não
(a|b|c|d)e	não	sim	sim	não	não	não
(a|b)*c						
(a|b)+c						

Agrupamento - exemplo

	abcde	de	be	c	abc	ababc
(abc)?de	sim	sim	não	não	não	não
(a|b|c|d)e	não	sim	sim	não	não	não
(a|b)*c	não	não	não	sim	sim	sim
(a|b)+c						

Agrupamento - exemplo

	abcde	de	be	c	abc	ababc
(abc)?de	sim	sim	não	não	não	não
(a|b|c|d)e	não	sim	sim	não	não	não
(a|b)*c	não	não	não	sim	sim	sim
(a|b)+c	não	não	não	não	sim	sim

Outros

```
.         # Qualquer caracter
[aB9]     # a OU B OU 9
[0-9]     # Qualquer caractere
           numérico
[a-zA-Z]  # Qualquer letra
[^a-c]    # Qualquer caractere
            exceto a, b OU C
```

Outros

```
\d    # Como [0-9]

\w    # Como [a-zA-Z0-9]

\W    # Como [^a-zA-Z0-9]

\s    # Como [ \t\n\r\f\v]
```

(espaço em branco, quebra de linha)

Outros

```
^          #  Início de string
$          #  Final de string
```

Flags

```
re.IGNORECASE | re.I

# Ignora maiúsculas e
minúsculas (Case sensitive)
```

Flags

```
re.MULTILINE | re.M

# '^' matches at the beginning
of each line
```

Flags

re.**DOTALL** |re.**S**

Faz com que o metacaractere . case com qualquer caractere, inclusive a quebra de linha *\n*. Sem esta opção, . casa qualquer caractere exceto o *\n*.

Flags

re.UNICODE |re.**U**

Faz os metacaracteres \w, \W, \b, \B, \d, \D, \s e \S também considerarem caracteres não-ASCII. Atributos definidos no módulo *unicodedata* definem caracteres Unicode que também pode ser considerados alfanuméricos (como letras acentuadas) e outros tipos de espaços.

Exercícios

https://pythex.org/

Exercícios

Qualquer número seguido de uma vogal

http://dontpad.com/QUESTAORE1

Exercícios

Qualquer número seguido de uma vogal

R: (\d)*(a|e|i|o|u) IGNORECASE

Exercícios

Uma expressão regular que casa um CPF
ex:772.843.809-34

http://dontpad.com/QUESTAORE2

Exercícios

Uma expressão regular que casa um CPF
ex:772.843.809-34

R: ^\d{3}\.\d{3}\.\d{3}\-\d{2}$

Exercícios

Escreva uma regex capaz de encontrar no texto todas as palavras terminam com vogais.

http://dontpad.com/QUESTAORE3

Exercícios

Escreva uma regex capaz de encontrar no texto todas as palavras terminam com vogais.

R: (\w)+(a|e|i|o|u)(|\.)

Funções no python para RE

```
re.search(pattern, string, flags=0)

re.match(pattern, string, flags=0)

re.split(pattern, string, maxsplit=0,
flags=0)

re.sub(pattern, repl, string, count=0,
flags=0)

re.findall(pattern, string, flags=0)

re.finditer(pattern, string, flags=0)

re.subn(pattern, repl, string,
count=0, flags=0)

re.escape(pattern)
```

Search vs. Match

Os métodos de *match* e *search* tomam uma string como argumento e devolvem um objeto *Match* com informações sobre o padrão encontrado ou *None* caso o padrão não seja encontrado.

Match

O método match verifica se a expressão regular casa com o texto desde o início

Search

O método de *search* percorre o texto para tentar encontrar um casamento.

Split

Divide a string pela ocorrência de um padrão

Sub

Retorna a string obtida, substituindo um padrão por um parâmetro. Se o padrão não é encontrado, a string é retornada sem mudanças.

Findall

Retorna os valores casados em forma lista, se não encontrar nada retorna uma lista vazia.

Exercício

Usando o dataset do kaggle disponível em notebook practical_example.ipynb, recupere subject e o body dos e-mails.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis at vitae vestibulum sem, sed tempus sem. Integer eget diam metus. Cras at Muito Obrigado! orci. Suspendisse porta at turpis nec mauris vestibulum, est a eleifend est bibendum. posuere odio vel orci tempor venenatis. Vivamus eleifend, eros ac ultrices porta, diam Lorem elit felis est rutrum, dalai.ribeiro@gmail.com ante ut amet vel romulocosta100@gmail.com ligula. at Lorem ipsum dolor sit amet, consectetur vel at.