Mineração de Textos – Weka e Python

Profa Daniela Souza Moreira da Silva

30 ANOS
CIÊNCIA DA COMPUTAÇÃO
GRANDE FLORIANÓPOLIS
UNIVALI

Mineração de Textos

- É baseada na **busca por padrões** em **textos digitais em linguagem natural**.
- Texto digital visto como uma das principais fontes de informação.
- Dados textuais estão, de modo geral, não estruturados.
- Mineração de Dados x Mineração de Textos
- Etapas da Mineração de Textos

Pré - processamento → Processamento Textual → Pós - processamento

- Também chamada de PLN
- Coleta de dados textuais
- Normalização textual

Mineração de Textos

- O que linguagem natural?
  - É aquela que foi desenvolvida e evoluída por seres humanos a partir do uso natural do dom de se comunicar, isto é, não foi criada artificialmente como uma linguagem de programação (SARKAR, 2016).

Lematização
- Estudo
- Estudas
- Estudamos

Estudo (N)
Ou Estudar (V)

Radicalização
- Estudo
- Estudas
- Estudamos

Estud (radical)

- Etapas mais comuns da Normalização Textual:
  - Limpeza textual (Retirada de marcadores textuais, tags, comentários ..)
  - Uniformização maiúscula/minúsculas
  - Remoção de símbolos e pontuação
  - Tokenização do texto (unidade mínima, chamada de token, normalmente são palavras)
  - Expansão de contrações (tromba d'agua -> tromba de agua)
  - Remoção de stopwords (artigos, pronomes, advérbios, conjunções)
  - Correção da grafia
  - Lematização/ Radicalização (stemming)
    - Lematização: busca a forma canônica de um conjunto de palavras (Será classificada como verbo ou substantivo.
    - Radicalização: busca o radical em comum de um conjunto de palavras.

Sobre WEKA

- WEKA (**W**aikato **E**nvironment for **K**nowledge **A**nalysis) é um software para mineração de dados, desenvolvido pela Universidade de Waikato, Nova Zelândia.

- Weka é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Ele contém ferramentas para preparação de dados, classificação, regressão, agrupamento, mineração de regras de associação e visualização

- Link para download: https://waikato.github.io/weka-wiki/downloading_weka/

Sobre WEKA

- WEKA implementado na linguagem de programação JAVA, com a característica principal de sua portabilidade funcionando em diferentes sistemas operacionais.

- É um software de código aberto sob a licença GPL (General Public License).

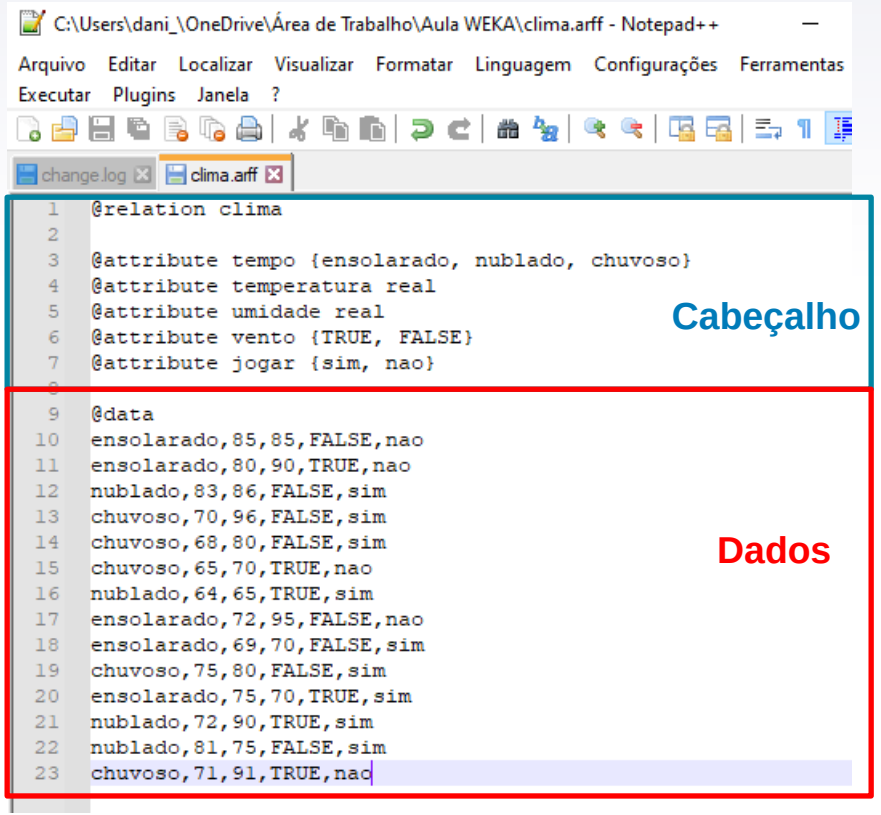
- Algoritmos (Métodos) implementados no WEKA:
  - Métodos de classificação (SVM, Árvore de Decisão,Naive Bayes, Regressão Lógica...)
  - Métodos de predição numérica (Regressão Linear,Percepton multicamadas...)
  - Métodos de agrupamento (SimpleKMeans, Clope, Cobweb...)
  - Métodos de associação (Apriori, FPGrowth...)

Sobre WEKA

- Possui vários algoritmos com a finalidade de realizar o pré-processamento dos dados.
- Os dados precisam estar organizados para realizar a mineração.
- Os arquivos podem estar em um formato específico, planilha ou banco de dados.
- O WEKA possui um formato para a organização dos dados ARFF(Attribute-Relation File Format).

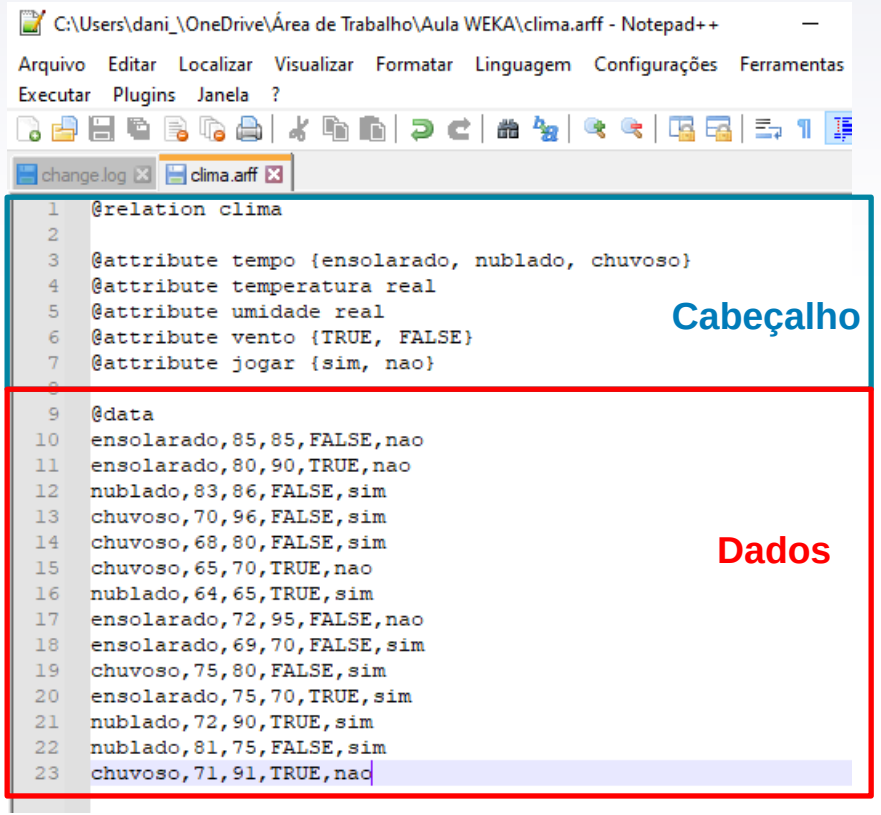
Arquivo ARFF (*Attribute-Relation File Format*)

- Arquivo dividido entre 2 seções: Cabeçalho e Dados.
- Cabeçalho:
  - Primeira linha deve apresentar a relação do arquivo "@relation "
  - Se o nome dessa relação contiver espaços deve estar entre aspas
  - Os atributos devem ser únicos, terão um nome e um tipo precedidos da palavra reservada @attribute.
    - O nome do atributo deve começar com letra e, se contiver espaços, deve estar entre aspas
    - Tipos de Dados:
      - Números(reais ou inteiros): Numeric ou Real
      - Texto "livre": String
      - Atributos categóricos(lista de valores)
      - Data: Date [<date-format>]



Arquivo ARFF (*Attribute-Relation File Format*)

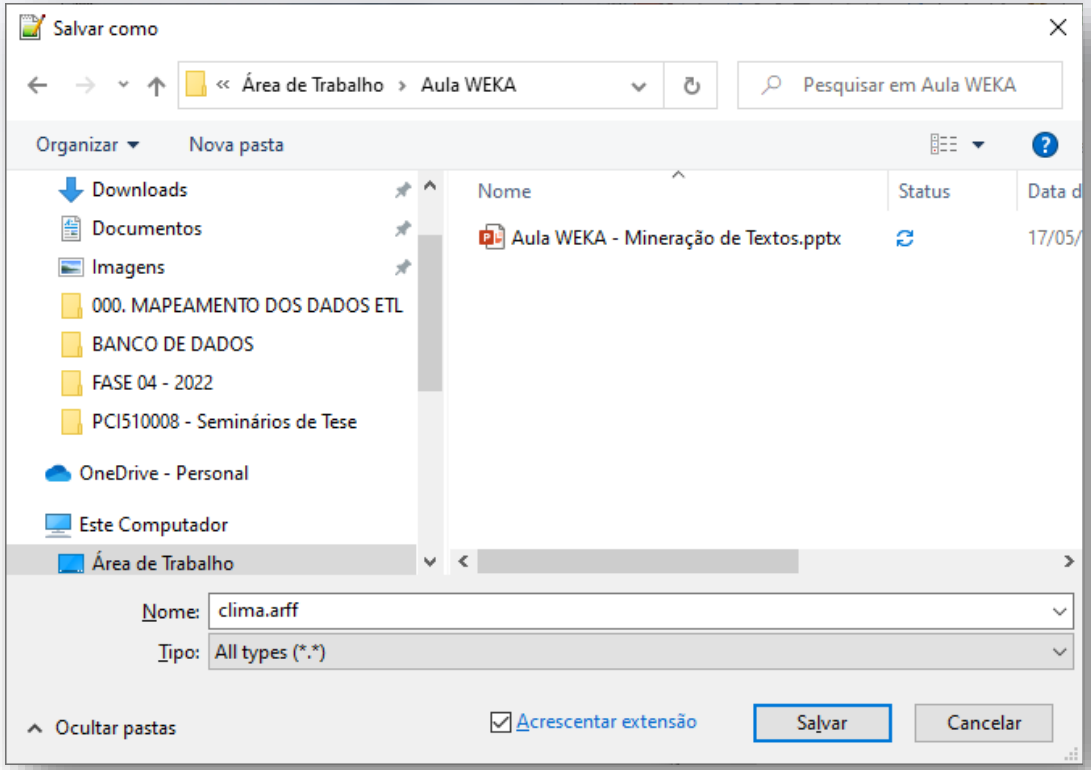
- Arquivo dividido entre 2 seções: Cabeçalho e Dados.
- Dados:
  - A base de dados vem logo após o @data
  - É a lista de todas as instancias com os valores dos atributos separados por vírgulas. Cada instância (registro) é representada em uma única linha.
  - Os atributos devem aparecer na ordem em que são declarados no cabeçalho.
  - Por default, o WEKA trata o último atributo especificado no cabeçalho como o atributo classe e os demais como atributos preditivos.
  - No caso de dados textuais, na área de dados, os valores devem estar entre aspas.



Arquivo ARFF (*Attribute-Relation File Format*)

- Para gerar um arquivo arff utilizar o notepad++, e salvar o arquivo com o nome.arff, e o tipo (all types (*.*)

- Outra opção é o notepad, salvando o arquivo com o nome.arff e o tipo “Todos os arquivos (*.*)

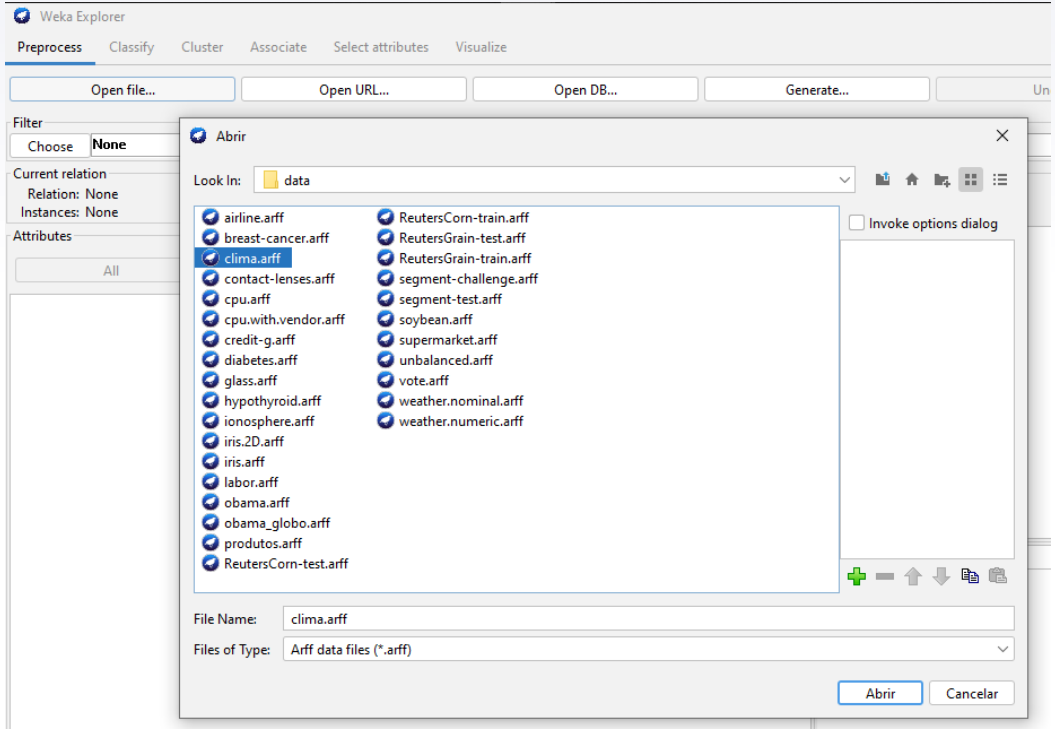


Weka

- A guia Explorer será utilizada para as atividades de mineração (dados ou textos).

Weka

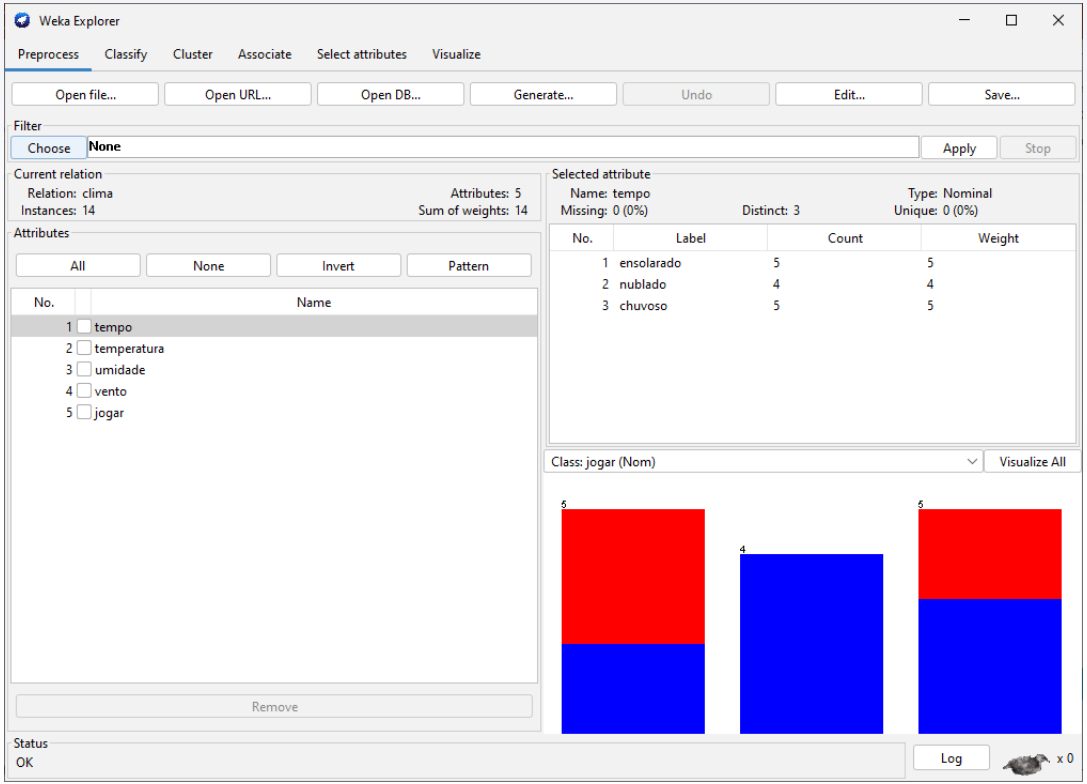
- Após clicar no Explorer, clicar na opção “Open File” e selecione o arquivo desejado (na extensão arff) e o programa carregará os dados na interface.



Weka

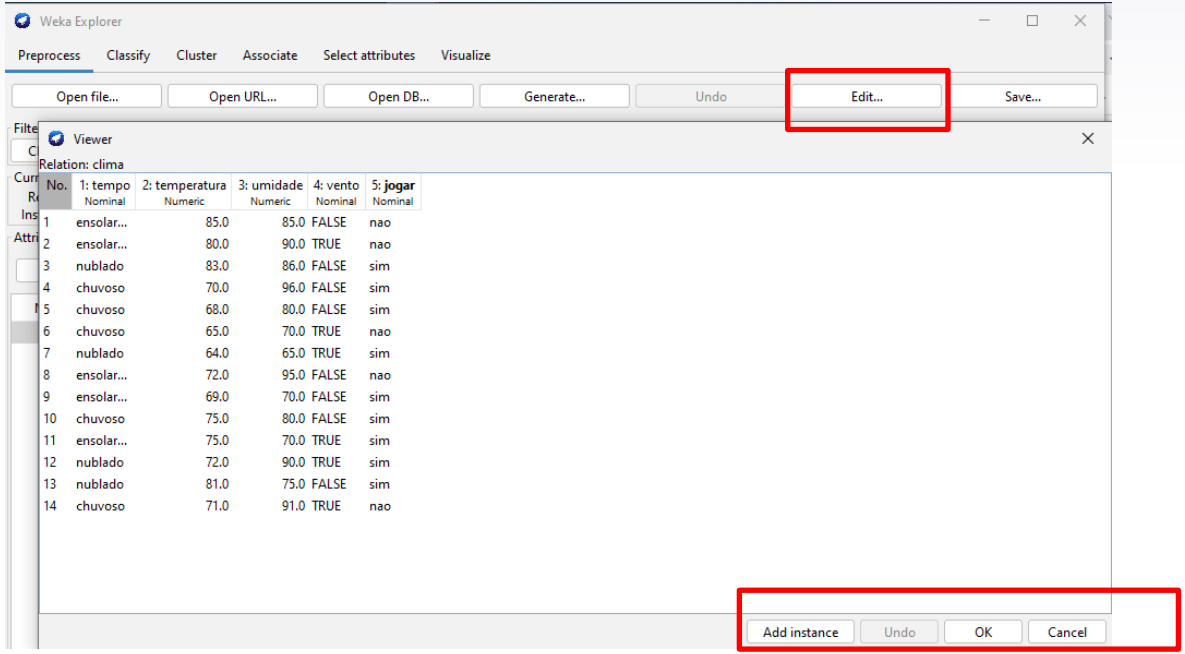
- Na aba “Preprocess” é onde será realizado o pré-processamento dos dados por meio dos filtros(Filter->Choose).

- Em “Current relation” pode-se ver o nome do arquivo(Relation), a quantidade de atributos(Attributes) e de instâncias(Instances)..



Weka

- Na opção “Edit” pode ser editado o arquivo de dados.

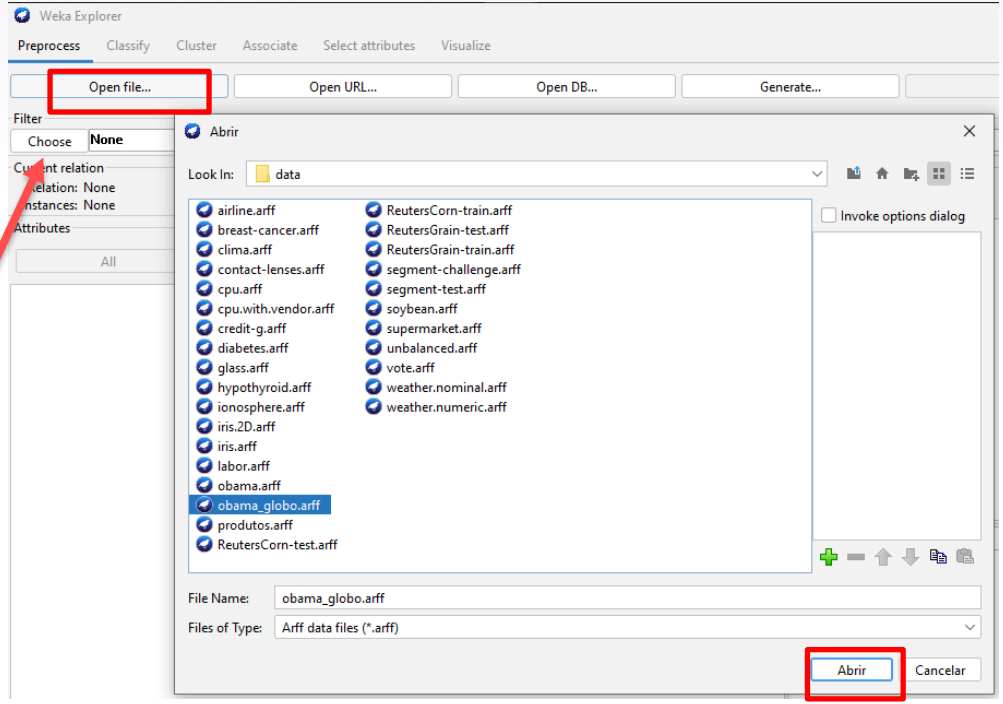


Mineração de Textos utilizando StringToVector do Weka

- Análise de textos utilizando WEKA: discurso do ex-presidente dos USA Barack Obama, que ocorreu no Theatro Municipal da cidade do Rio de Janeiro no dia 20/03/2011 (discurso disponível no G1).
- **Objetivo**: Minerar o texto para verificar o número de ocorrências das palavras para identificar qual foi o enfoque do discurso.
- **Metodologia**: Normalizar o texto (remoção de stopwords e uniformização maiúscula/minúscula) e utilizar um algoritmo que fracione o texto em palavras.
  - **Algoritmo**: StringToWordVector . Cria um vetor de palavras separando o texto.
  - Ajustar os parâmetros.

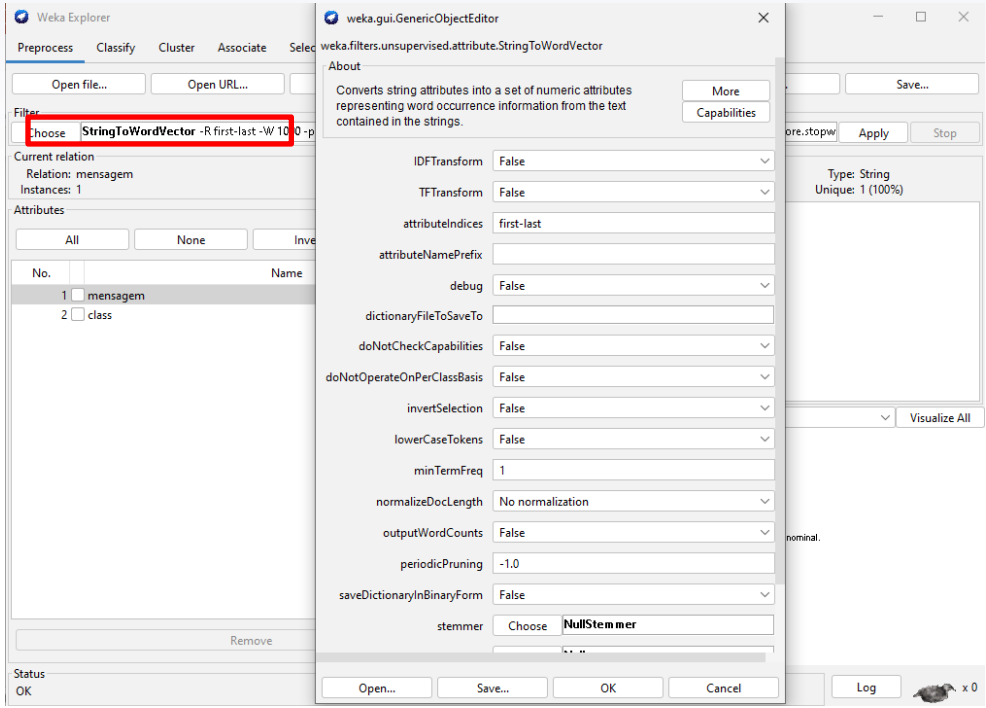
Mineração de Textos utilizando StringToVector do Weka

- 1º passo) Criar o arquivo arff com o discurso;
- 2º passo) Carregar o arquivo no WEKA, clicando em Open File, selecionando o diretório onde o arquivo foi salvo.
- 3º passo) Após abrir o arquivo, acessar a opção "Filter/Choose", e escolher a opção "filters->unsupervised->attribute->StringToWordVector".



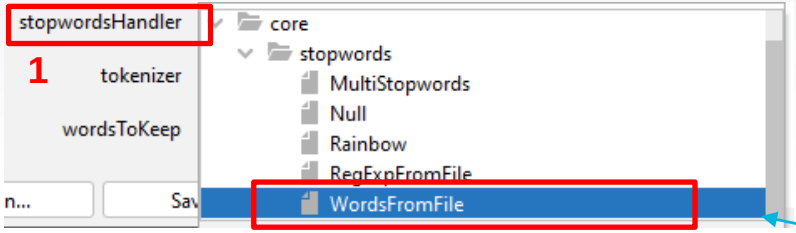
Mineração de Textos utilizando StringToVector do Weka

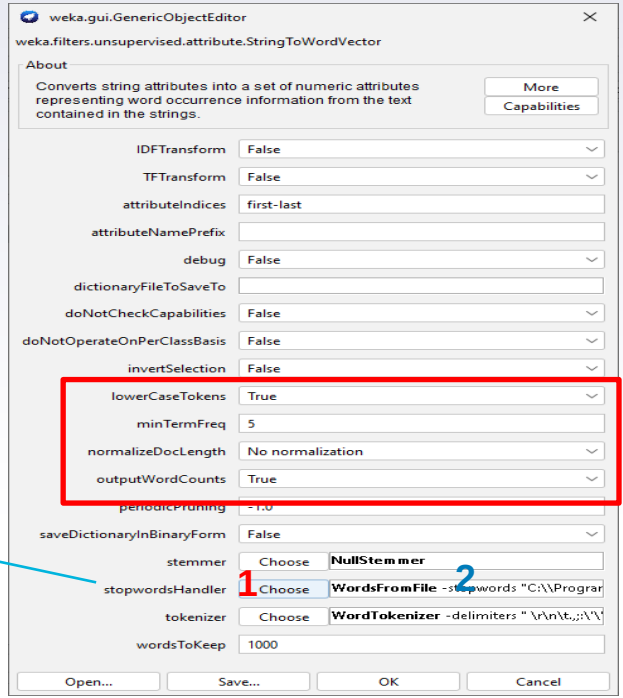
- 4º passo) Para ajustar os parâmetros do algoritmo, clicar na guia choose do filter e abrirá uma janela para configuração:
  - **LowerCaseTokens**: Transforma todas as palavras para minúscula
  - **MinTermFreq:** Foi estabelecido o valor mínimo de 5 palavras por ocorrência, a depender do texto a média de palavras por ocorrência pode ser aumentada.
  - **OutputWordCounts:** Conta a quantidade de ocorrências de cada palavra.
  - **WordsToKeep:** Estabelece o número máximo de palavras a serem analisadas.



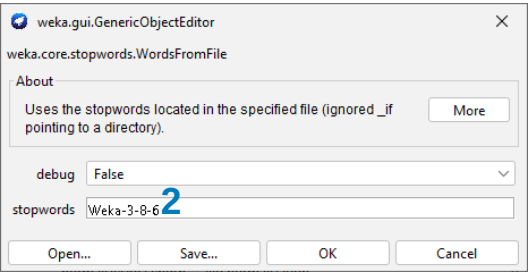
Mineração de Textos utilizando StringToVector do Weka

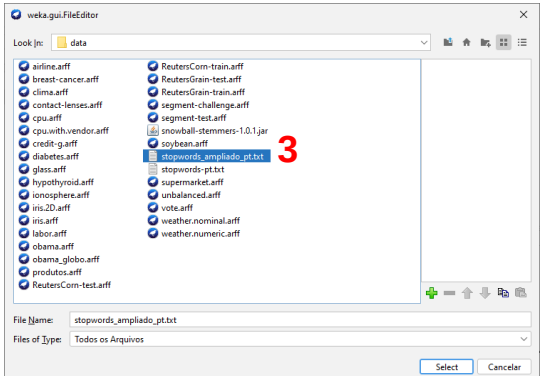
- 5º passo) Para utilizar o arquivo das stopwords, clicar na opção stopwordsHandler/Choose/WordsFromFile (ler palavras do arquivo)

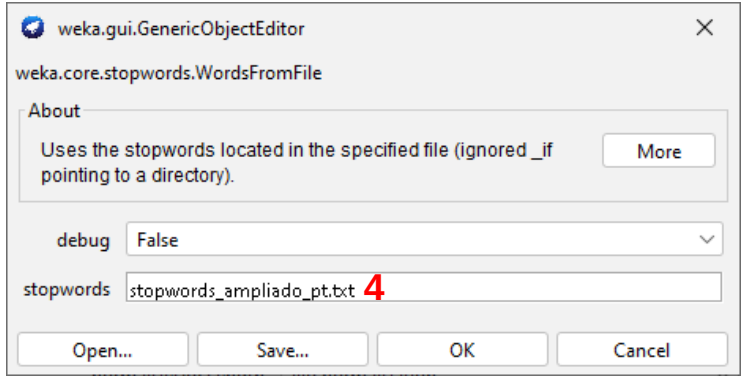


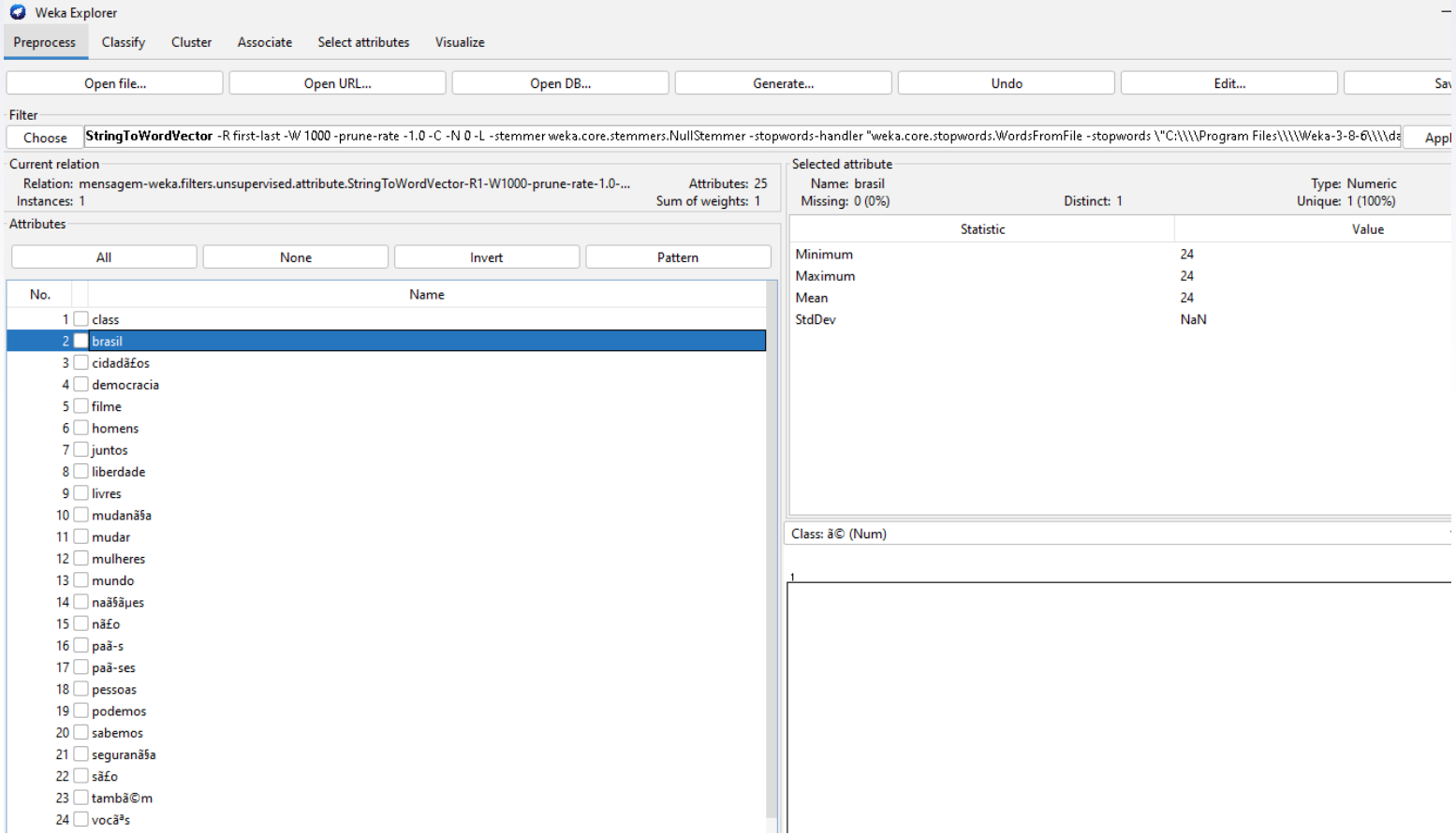


- 6º passo) Clicar na área wordFromFile e indicar o local onde está o arquivo com as stopwords. Coloca-lo na pasta data do WEKA.



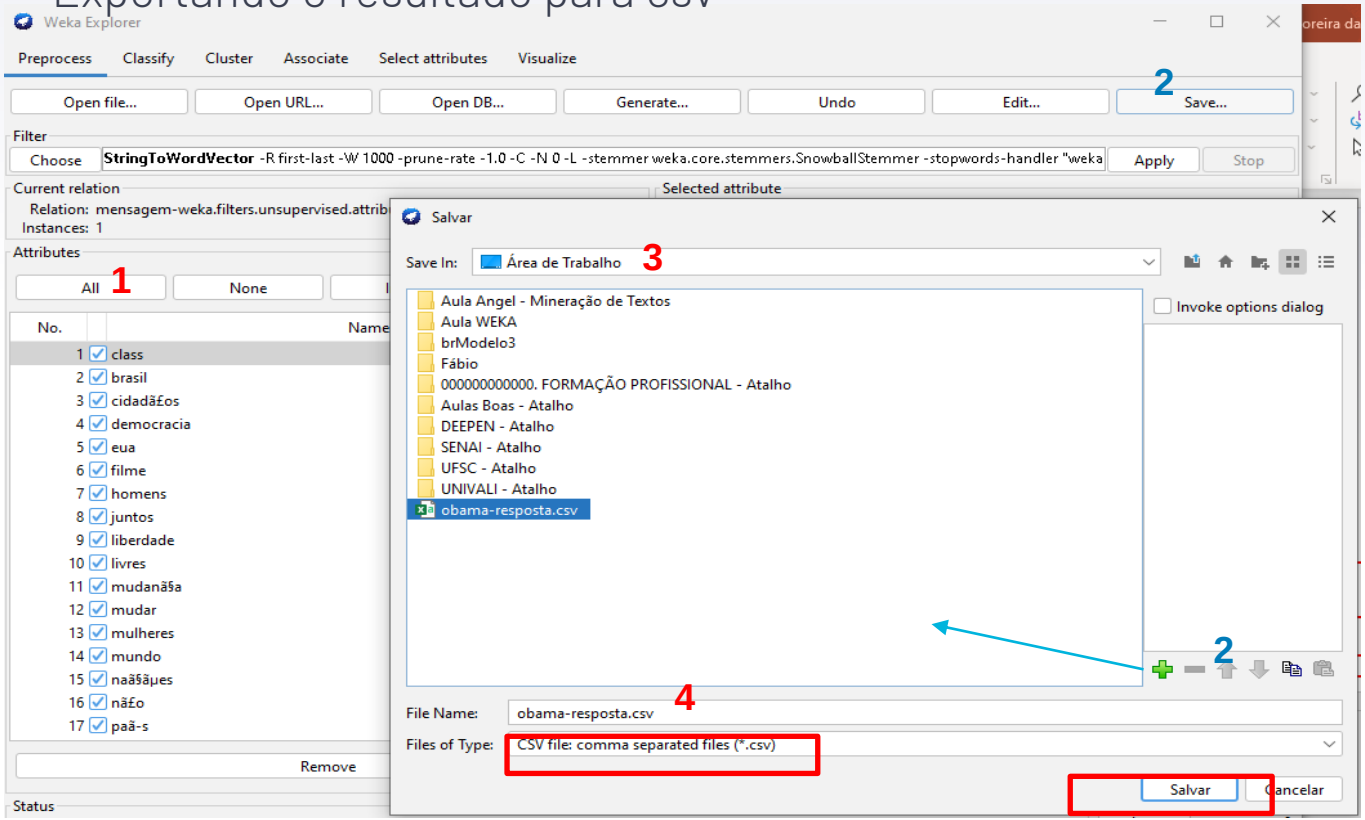





Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Filter
Choose
StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -C -N 0 -L -stemmer weka.core.stemmers.NullStemmer -stopwords-handler "weka.core.stopwords.WordsFromFile -stopwords \"C:\\\\Program Files\\\\Weka-3-8-6\\\\da
Current relation
Relation: mensagem-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-...
Instances: 1
Attributes: 25
Sum of weights: 1
Selected attribute
Name: brasil
Missing: 0 (0%)
Distinct: 1
Type: Numeric
Unique: 1 (100%)
Statistic
Value
Minimum 24
Maximum 24
Mean 24
StdDev NaN
Attributes
All
None
Invert
Pattern
No.
Name
1 class
2 brasil
3 cidadãƒos
4 democracia
5 filme
6 homens
7 juntos
8 liberdade
9 livres
10 mudanãša
11 mudar
12 mulheres
13 mundo
14 naãšãµes
15 nãƒo
16 paã-s
17 paã-ses
18 pessoas
19 podemos
20 sabemos
21 seguranãša
22 sãƒo
23 tambã©m
24 vocãªs
Class: ã© (Num)
1

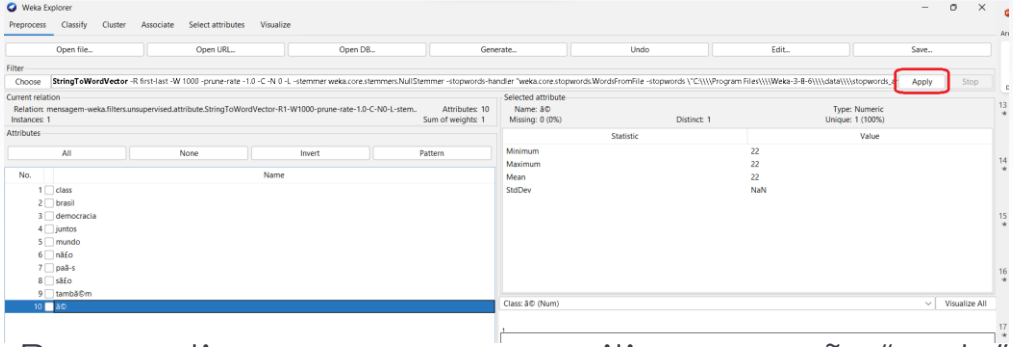
Mineração de Textos utilizando StringToVector do Weka

- Exportando o resultado para csv

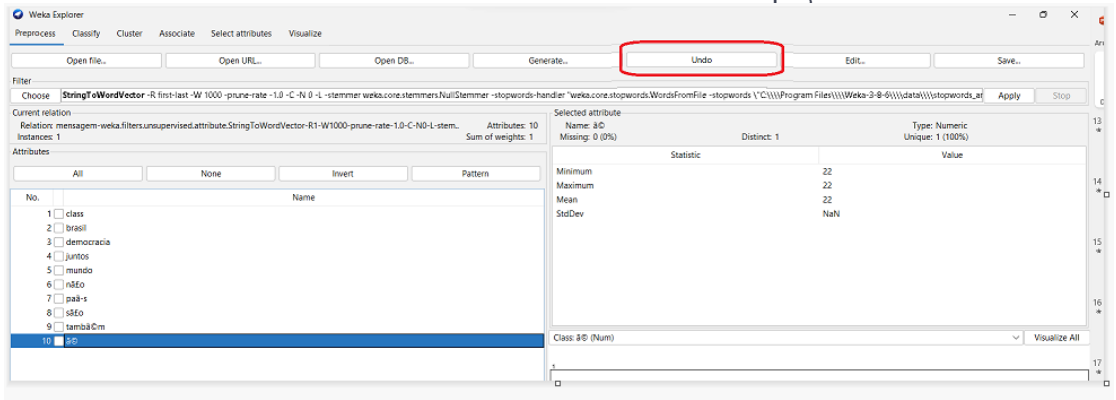


Executando os testes

- Após realizar a parametrização do filtro, clicar em “Apply” para rodar o algoritmo.

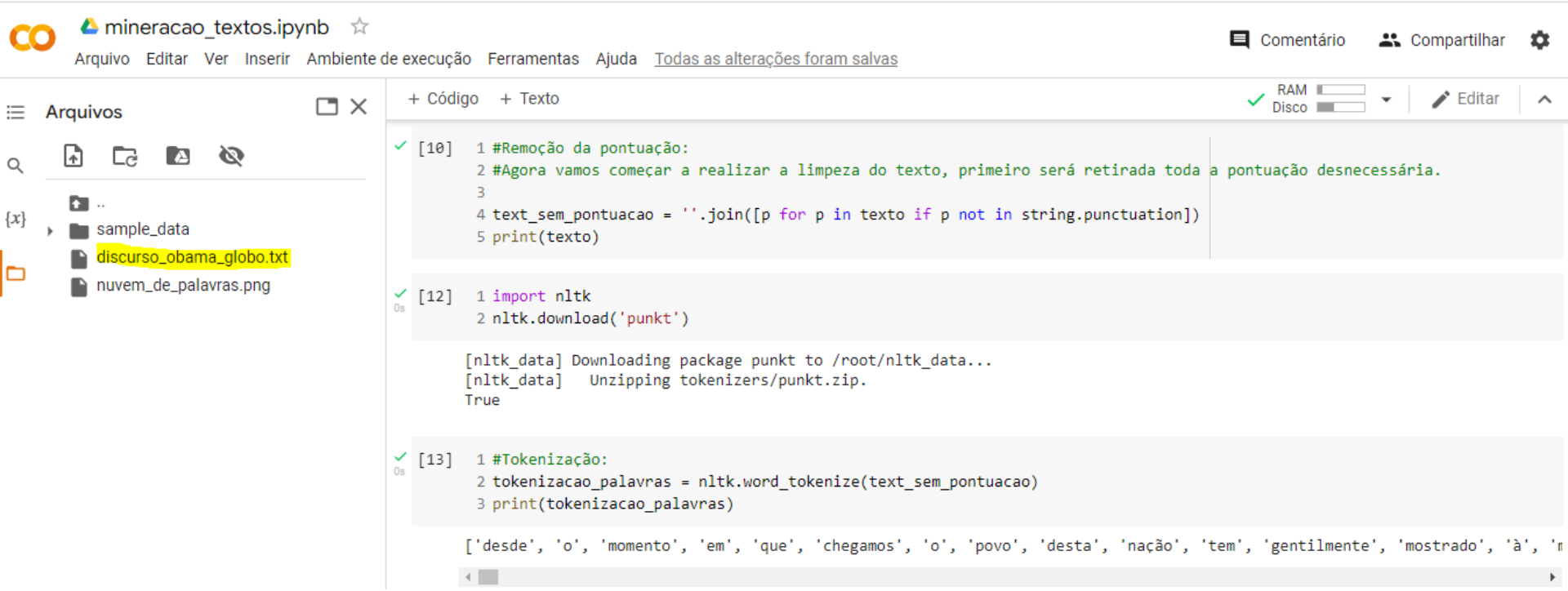


- Para realizar novos testes utilizar a opção “undo”. Ela remove as configurações



Mineração de Textos com Python

Mineração de Textos com Python



Mineração de Textos com Python

mineracao_textos.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Comentário Compartilhar

Arquivos

..
sample_data
discurso_obama_globo.txt
nuvem_de_palavras.png

+ Código + Texto

RAM Disco

Editar

```
[14]  1 #Remoção das Stopwords:
      2
      3 nltk.download('stopwords')
      4 stopwords = stopwords.words('portuguese')
      5
      6 palavras_sem_stop = [p for p in tokenizacao_palavras if p not in stopwords]
      7 print(stopwords)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
['de', 'a', 'o', 'que', 'e', 'é', 'do', 'da', 'em', 'um', 'para', 'com', 'não', 'uma', 'os', 'no', 'se', 'na', 'por', 'mais
```

```
[15]  1 #Verificando as palavras com mais frequência:
      2 #A função .most_common() da biblioteca nltk irá retornar as 10 palavras que mais aparecem no texto
      3
      4 freq = FreqDist (palavras_sem_stop)
      5 freq = freq.most_common(40)
      6 print(freq)
```

```
[('brasil', 24), ('país', 18), ('eua', 13), ('aqui', 12), ('mundo', 12), ('ser', 12), ('povo', 10), ('hoje', 10), ('juntos'
```

Mineração de Textos com Python

mineracao_textos.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

Comentário Compartilhar

Arquivos

..
sample_data
discurso_obama_globo.txt
nuvem_de_palavras.png

+ Código + Texto

RAM Disco

Editar

[16]
```
#Gerando a wordcloud:
nuvem_palavras = WordCloud(background_color='white',stopwords = stopwords,height=1080,width=1080,max_words = 100)
nuvem_palavras.generate(texto)
nuvem_palavras.to_file('nuvem_de_palavras.png')

```

```
<wordcloud.wordcloud.WordCloud at 0x7ff901716cd0>
```

```
from nltk.stem import SnowballStemmer

stem = SnowballStemmer('portuguese')
for p in palavras_sem_stop:
  print(stem.stem(p))
```

```
cheg
aqu
agor
hor
tom
poss
país
sempr
concord
tud
```

Mineração de Textos com Python

Referências

- https://www.cs.waikato.ac.nz/ml/weka/index.html
- https://www.ranks.nl/stopwords/portuguese
- https://g1.globo.com/obama-no-brasil/noticia/2011/03/leia-integra-do-discurso-de-barack-obama-no-theatro-municipal.html
- https://www.ufsm.br/pet/sistemas-de-informacao/2021/07/12/introducao-a-mineracao-de-textos-com-python/
- https://github.com/Prof-Rodrigo-Silva/Text-Mining
- https://www.linkedin.com/pulse/minera%C3%A7%C3%A3o-de-texto-usando-stringtowordvector-do-weka-rosa-da-silva/

Referências

- https://code.google.com/archive/p/ptstemmer/

Obrigada!

Perguntas?

Dani.smoreira@gmail.com

Slides Carnival

Free templates for all your presentation needs

For PowerPoint and Google Slides

100% free for personal or commercial use

Ready to use, professional and customizable

Blow your audience away with attractive visuals