

ANÁLISE DE SENTIMENTOS UTILIZANDO TÉCNICAS DE CLASSIFICAÇÃO MULTICLASSE

Utilizando Técnicas de Extração de Características e Algoritmos de
Aprendizado de Máquina para a Classificação Binária e Multiclasse

Alexandre Lunardi
José Viterbo
Flávia Bernardini

SUMÁRIO

- Introdução a Análise de Sentimentos
- Introdução ao Python
- Análise de Sentimentos Multiclasse
- Extração de Características
- Algoritmos de Aprendizado
- Conclusões e Trabalhos Relacionados

INTRODUÇÃO

- Web 2.0 e Web Social;
- Redes sociais e blogs;
- Sistemas e-commerce;
- Recuperação e mineração de dados.



MOTIVAÇÃO

- Alugar filmes - IMDb;
- Comprar algum produto;
- Reservar hotéis - TripAdvisor;
- Política - Twitter.

68 of 123 people found the following review helpful

★☆☆☆☆ **Disappointing**, August 13, 2012

By [AvidReader](#)

Amazon Verified Purchase ([What's this?](#))

This review is from: Where We Belong (Kindle Edition)

This book was so disappointing. I have read all of Emily Giffin's books, and have found that her last few books are getting worse and worse. Where We Belong had the ability to be a great story. However, telling the story from two points of view, Marianne and Kirby, led there to be little depth to either character. Also I found both characters to be very unlikable. The story was trite and unbelievable. I also found that Giffin put a very negative spin on adoption. Giffin's last books have been a disappointment and this one was no different.

Help other customers find the most helpful reviews

[Report abuse](#) | [Permalink](#)

Was this review helpful to you?

Sentiment Analysis with Python NLTK Text Classification

This is a demonstration of **sentiment analysis** using a **NLTK 2.0.4** powered **text classification** process. It can tell you whether it thinks the text you enter below expresses **positive sentiment**, **negative sentiment**, or if it's neutral. Using **hierarchical classification**, *neutrality* is determined first, and *sentiment polarity* is determined second, but only if the text is not neutral.

Analyze Sentiment

Language

english ▼

Enter text

the lord of the rings is the best movie

Enter up to 50000 characters

Analyze

Sentiment Analysis Results

The text is **pos**.

The final sentiment is determined by looking at the classification probabilities below.

Subjectivity

- neutral: 0.3
- polar: 0.7

Polarity

- pos: 0.9
- neg: 0.1

SENTIMENT140

Sentiment140 [Tweet](#)

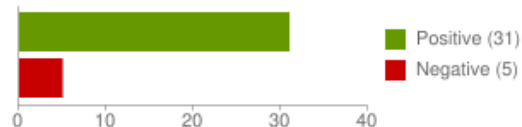
English ▾Search

Sentiment analysis for patriots

Sentiment by Percent



Sentiment by Count



Tweets about: patriots

[parmiegvel](#): NFL Power Rankings - Week 4: **Patriots** maintain No. 1 spot, Falcons surge into top 10 <http://t.co/JGyveRTM3O> (via <http://t.co/KljCJVyUfG>)

Posted: 21 seconds ago

[Anthony's_Era](#): RT [@NEPD_Loyko](#): I'm sure **#Patriots** pass rush could be that much better if BB allowed them free reign to rush and get after the QB.. but lik?

Posted: 46 seconds ago

OBJETIVOS

- Apresentar o conceito de análise de sentimentos;
- Apresentar as Técnicas de Extração de Características;
- Apresentar os algoritmos de Aprendizado;
- Realizar um estudo de caso.

A ANÁLISE DE SENTIMENTOS

“Capturar e processar opiniões a fim de auxiliar um usuário ou uma empresa”

[Cambria et al., 2013]

DEFINIÇÕES

- Segundo [Liu, 2012], uma opinião regular, dada por um usuário, é representada como uma quintupla $O = (e, a, s, h, t)$

68 of 123 people found the following review helpful

★☆☆☆☆ **Disappointing**, August 13, 2012

By [AvidReader](#)

Amazon Verified Purchase ([What's this?](#))

This review is from: Where We Belong (Kindle Edition)

This book was so disappointing. I have read all of Emily Giffin's books, and have found that her last few books are getting worse and worse. Where We Belong had the ability to be a great story. However, telling the story from two points of view, Marianne and Kirby, led there to be little depth to either character. Also I found both characters to be very unlikable. The story was trite and unbelievable. I also found that Giffin put a very negative spin on adoption. Giffin's last books have been a disappointment and this one was no different.

Help other customers find the most helpful reviews

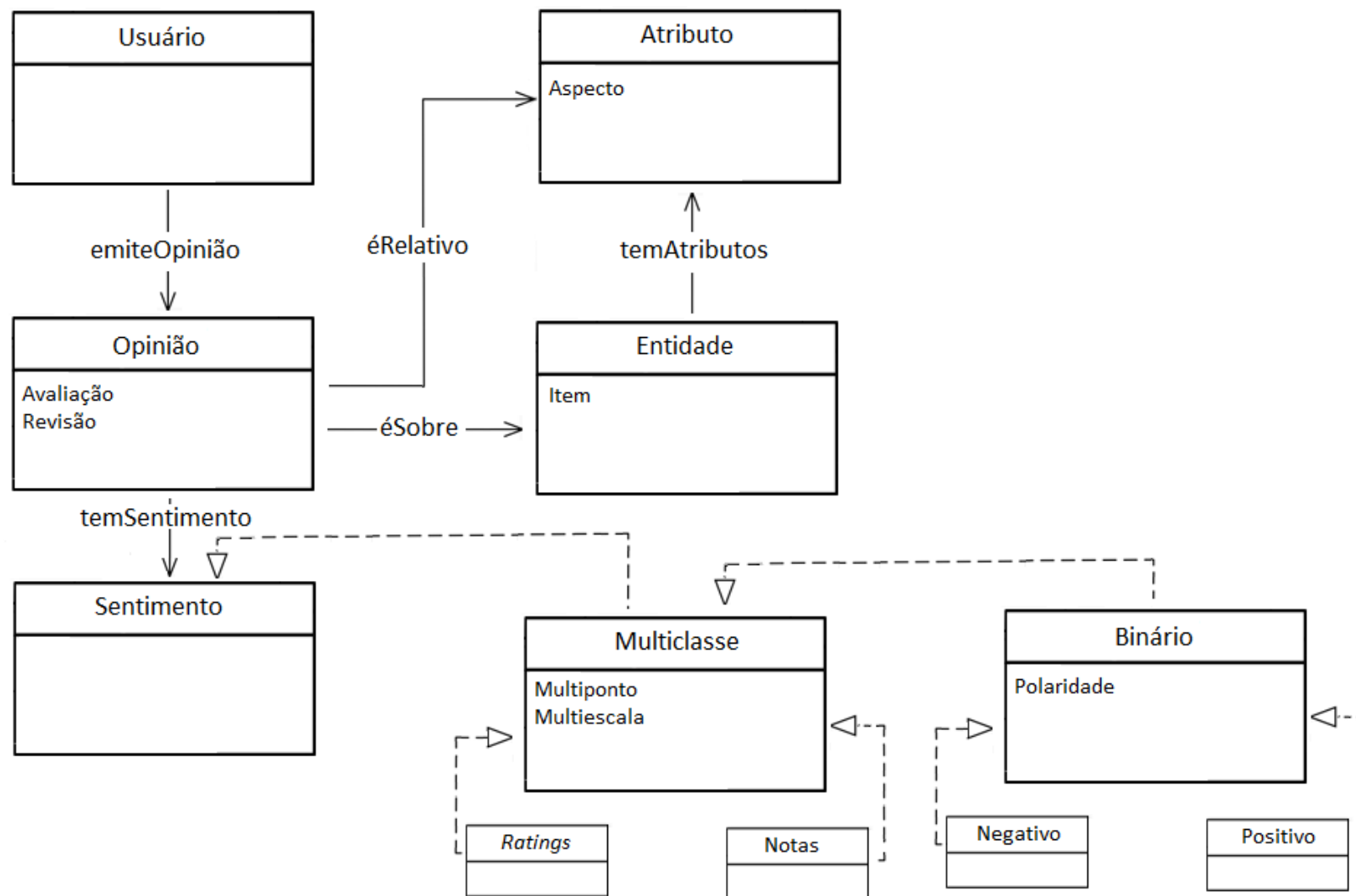
[Report abuse](#) | [Permalink](#)

Was this review helpful to you?

CLASSIFICAÇÃO BINÁRIA E MULTICLASSE



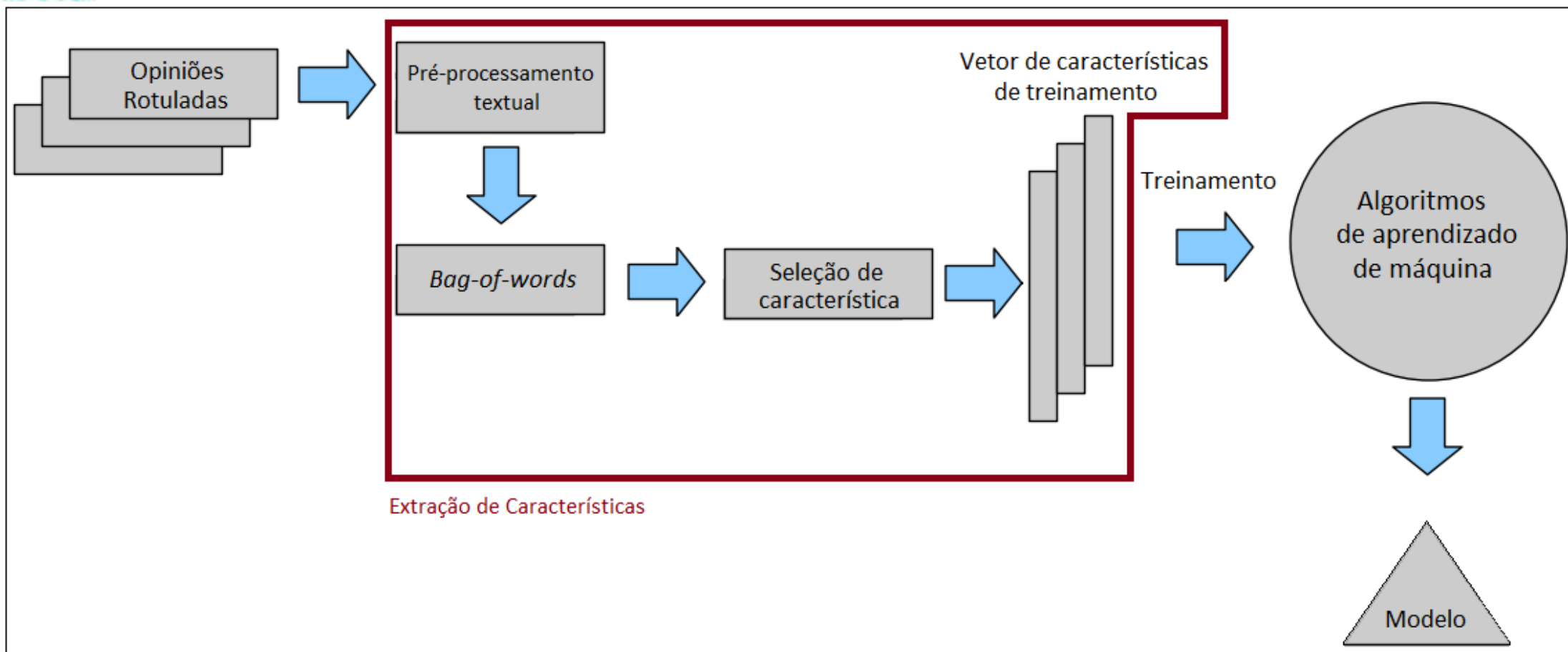
ONTOLOGIA – ANÁLISE DE SENTIMENTOS



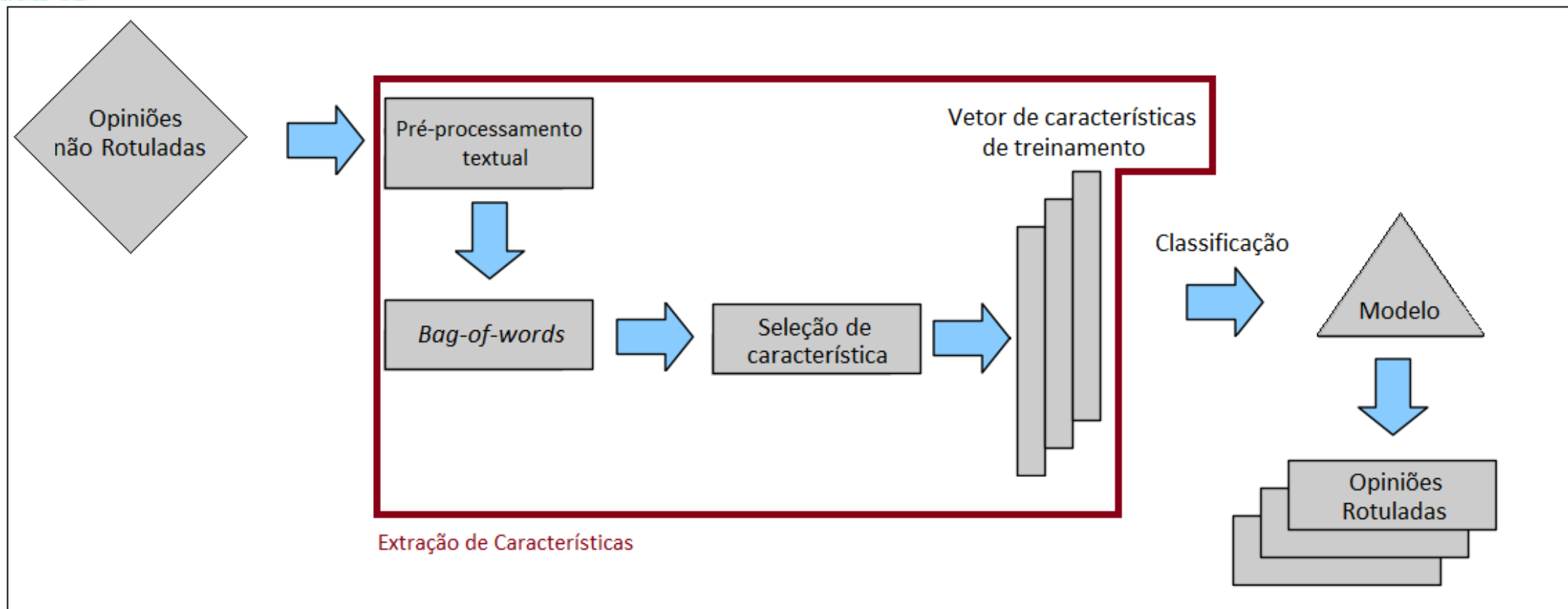
FORMAS DE REALIZAR A ANÁLISE DE SENTIMENTOS

- Aprendizado de máquina;
- Análise léxica;
- Orientação semântica;
- Análise conceitual.

APRENDIZADO DE MÁQUINA: TREINAMENTO



APRENDIZADO DE MÁQUINA: CLASSIFICAÇÃO



REVISÃO DA LITERATURA

Pré-processamento textual	<i>Bag-of-words</i>	Seleção de características	Vetorização	Algoritmos
Retirada de caracteres especiais	Unigrama	Chi quadrado	Frequência	Naive Bayes
Retirada de <i>stopwords</i>		Ganho de Informação		SVM - OvO
Tratamento da negação	Unigramas+ Bigramas	Ganho Médio	<i>tfidf</i>	SVM – OvA
				kNN
				Árvores de Decisão

INTRODUÇÃO AO PYTHON

- Linguagem interpretada;
- Código-fonte aberto;
- Disponível para vários S.O.;
- Orientada a objetos;
- Códigos com o símbolo `>>>` ou `[n]`.
- Endereço:

<http://www.dcc.ufrj.br/~fabiom/mab225/pythonbasico.pdf>

POR QUE PYTHON?

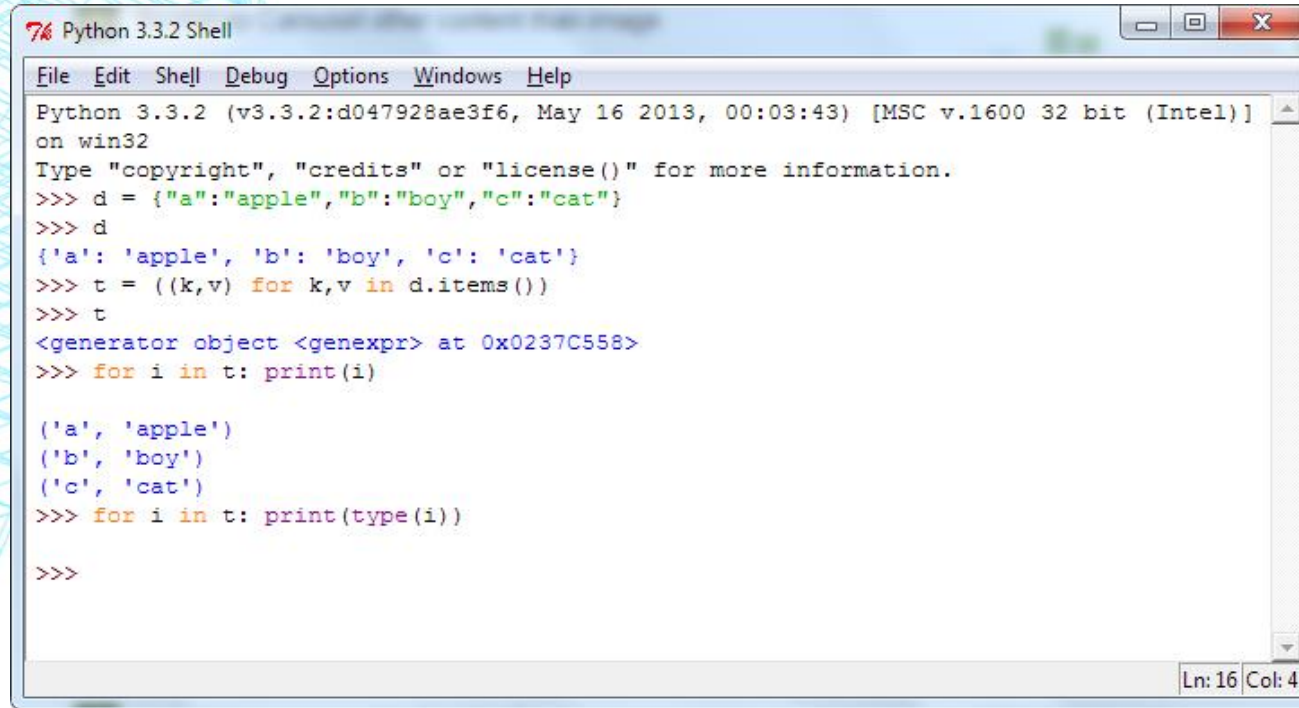
Quem usa Python?



<https://us.pycon.org/2013/sponsors/>



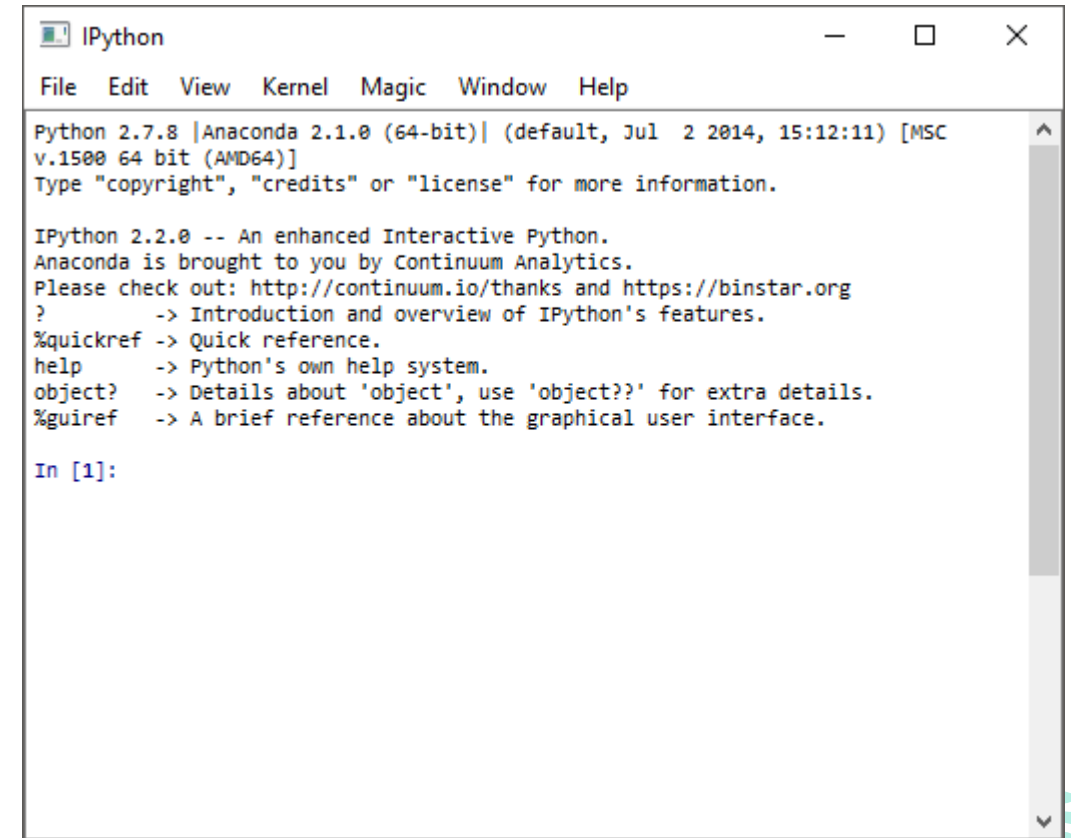
IDLE PYTHON



```
Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> d = {"a": "apple", "b": "boy", "c": "cat"}
>>> d
{'a': 'apple', 'b': 'boy', 'c': 'cat'}
>>> t = ((k,v) for k,v in d.items())
>>> t
<generator object <genexpr> at 0x0237C558>
>>> for i in t: print(i)

('a', 'apple')
('b', 'boy')
('c', 'cat')
>>> for i in t: print(type(i))

>>>
```



```
IPython
File Edit View Kernel Magic Window Help
Python 2.7.8 [Anaconda 2.1.0 (64-bit)] (default, Jul 2 2014, 15:12:11) [MSC
v.1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

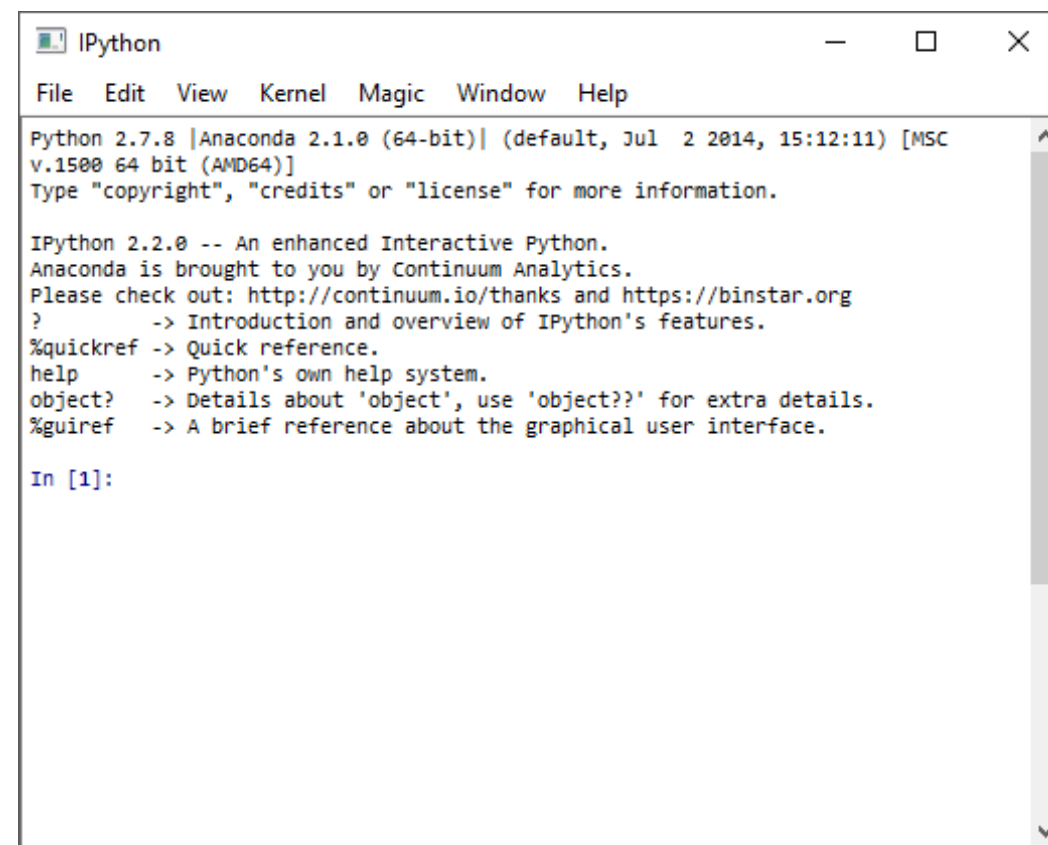
IPython 2.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui?   -> A brief reference about the graphical user interface.

In [1]:
```

Link para download: <https://dl.dropboxusercontent.com/u/70544691/minicurso.rar>
<https://dl.dropboxusercontent.com/u/70544691/Total.rar>
<https://dl.dropboxusercontent.com/u/70544691/Final.rar>
<https://dl.dropboxusercontent.com/u/70544691/exemplos.rar>

ANACONDA

- Distribuição livre com mais de 400 pacotes Python;
- Numpy, SciPy, SciKit...
- Disponível para Linux, OS e Windows
- Exemplo: Versão 2.7 possui 454 pacotes:
<https://docs.continuum.io/anaconda/pkg-docs>



```
IPython
File Edit View Kernel Magic Window Help
Python 2.7.8 [Anaconda 2.1.0 (64-bit)] (default, Jul 2 2014, 15:12:11) [MSC
v.1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 2.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%gui?   -> A brief reference about the graphical user interface.

In [1]:
```

VARIÁVEIS

- Três tipos básicos:
- Int
 1. `>>> a = 123`
 2. `>>> print a`
- Float
 1. `>>> b = 12.3`
 2. `>>> print b`
- String
 1. `>>> texto = 'Olá Mundo'`
 2. `>>> texto`
 3. `>>> print texto`
- Verificar tipo: `>>> type(variável)`

OBJETOS - STRINGS

- Uma *String* é uma sequência de letras:
 - `>>> texto[2]`
- Intervalo de sequência
 - `>>> texto [2:5]`
- Intervalo de sequência (último da sequência)
 - `>>> texto [2:]`
- Inverter uma sequência
 - `>>> texto[::-1]`

OBJETOS - STRINGS

- Concatenação

- `>>> texto = texto + 'para todos'`
- `>>> print texto`
- `>>> print texto[7:]`
- `>>> len(texto)`

- Formatação de Strings:

- `>>> pi = 3.14`
- `>>> print 'O valor de pi é %f' % pi`
- `>>> fruta = 'abacaxi'`
- `>>> print '%s é uma fruta' %fruta`

OBJETOS - LISTAS

- Uma lista também é uma sequência:
 - `>>> lista = [1,2,3]`
- Concatenação de listas:
 - `>>> lista = lista + [4]`
- Último endereço:
 - `>>> lista[-1]`
- Listas são sequências mutáveis:
 - `>>> lista[0] = 'zero'`

OBJETOS - LISTAS

- Aplicação de listas: matrizes
 - `>>> linha1 = [1,2,3]`
 - `>>> linha2 = [1,0,4]`
 - `>>> linha3 = [5,1,2]`
 - `>>> matriz = [linha1, linha2, linha3]`
 - `>>> matriz[1][2]`
- Manipulação de listas:
 - `>>> linha1.append('valor')`
 - `>>> linha1.extend([0,0,9])`
 - `>>> linha1.insert(0, 'Oi')`
 - `>>> linha1.remove(0)`
 - `>>> linha1.remove('Oi')`
 - `>>> linha1.pop(0)`

OBJETOS - DICIONÁRIOS

- São contêineres com sistema de endereçamento por chaves. Cada chave tem um valor atribuído:
 - `>>> curso = {'nome': 'análise de sentimentos', 'duração': '2', 'cidade': 'Florianópolis'}`
 - `>>> curso['nome']` ou `>>> print curso['nome']`
- Dicionários são mutáveis:
 - `>>> curso['duração'] = curso['duração'] + 2`
- Verificar chaves do dicionário ou perguntar se existe:
 - `>>> curso.keys()`
 - `>>> curso.has_key('estado')`

OBJETOS - DICIONÁRIOS

- Adicionando uma chave:
 - `>>> curso['estado'] = 'SC'`
- Lista contendo o par (chave, valor da chave):
 - `>>> curso.items()`

ESTRUTURAS DE CONTROLE DE FLUXO

WHILE

```
[35]: while b < 5:  
...:     print b  
...:     b = b+1
```

IF-ELIF-ELSE

```
In [38]: x = 'vapor'  
  
In [39]: if x == 'líquido':  
...:     print 'Menos de 100º C'  
...: elif x == 'vapor':  
...:     print 'Mais de 100º C'  
...: else:  
...:     print 'Menos de 0º C'  
...:  
Mais de 100º C
```

FOR

```
In [44]: a = ['João', 'Rafael', 'Douglas']  
  
In [45]: for i in a:  
...:     print i  
...:  
João  
Rafael  
Douglas
```

FUNÇÕES

- Úteis para executar um bloco de código;
- `def nome_da_funcao(parametro1,parametro2,...):`
operações sobre os n parâmetros
- Exemplo:
 - `>>>def f(x):`
`return x**2`
 - `>>> f(3)`

MANIPULAÇÃO DE ARQUIVOS

- Função **open()**;
- Sintaxe: **open('endereço/nome_do_arquivo.extensão', 'modo_de_abertura');**
- **Modos de abertura:**
 - **r**: somente para leitura;
 - **a**: somente para escrita, concatenando ao final do arquivo;
 - **w**: somente para escrita, reescrevendo o conteúdo anterior.

MANIPULAÇÃO DE ARQUIVOS

- Escrevendo em um arquivo **write()**:
 - `>>> arquivo.write('Testando o arquivo')`
- Fechar um arquivo **close()**
- **Exemplo:**
 - `>>> arquivo = open('teste.txt', 'r')`
 - `>>> arquivo.write('Testando o arquivo')`
 - `>>> arquivo.close()`

ENTRADA DE DADOS COM IDLE

- Teste com o programa `primeiro_programa.py`;

```
In [1]: cd Anaconda/  
C:\Users\Alexandre\Anaconda
```

```
In [2]: cd Examples/  
C:\Users\Alexandre\Anaconda\Examples
```

```
In [3]: cd MLiA_SourceCode/  
C:\Users\Alexandre\Anaconda\Examples\MLiA_SourceCode
```

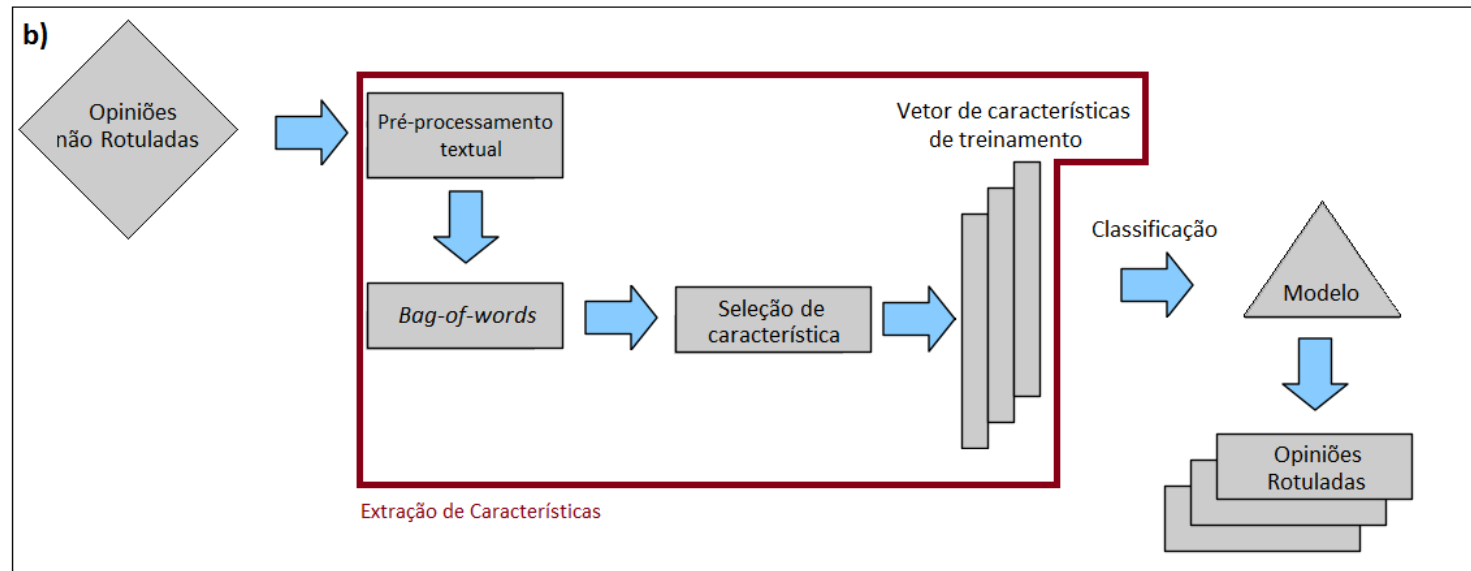
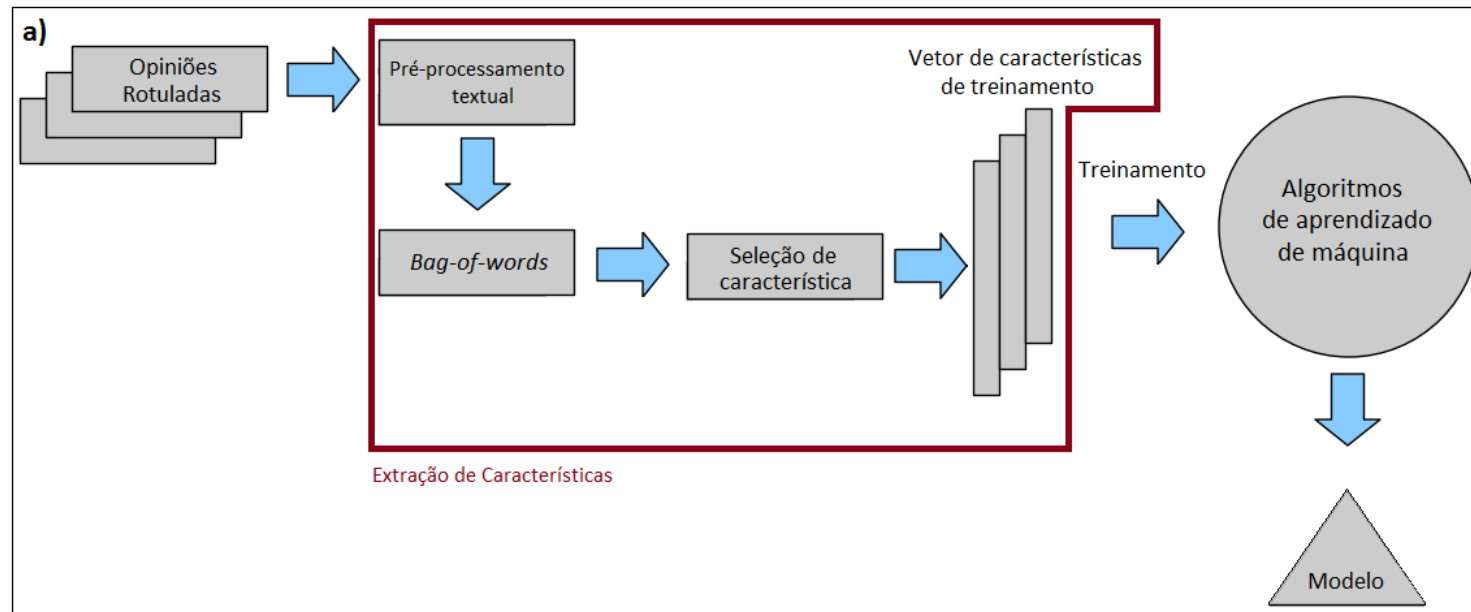
```
In [4]: cd minicurso/  
C:\Users\Alexandre\Anaconda\Examples\MLiA_SourceCode\minicurso
```

```
In [5]: import primeiro_programa
```

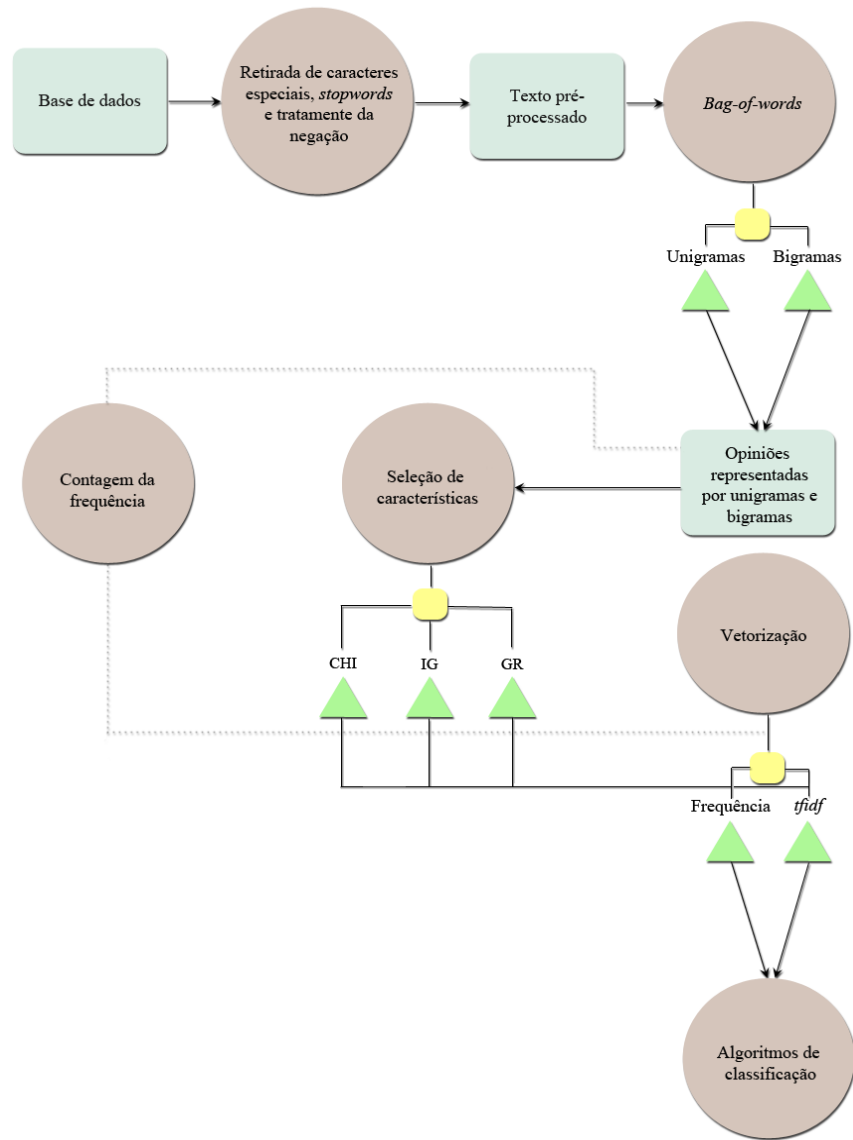
```
In [6]: primeiro_programa.testa_primo(11)  
Número primo
```

```
In [7]: primeiro_programa.testa_primo(12)  
Número não primo
```

RECAPITULANDO...



RECAPITULANDO...



ANÁLISE DE SENTIMENTOS MULTICLASSE

- Base de dados do TripAdvisor;
- Disponíveis em:

<http://times.cs.uiuc.edu/~wang296/Data/>



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN



ESTRUTURA DE UMA OPINIÃO

<Author>everywhereman2

<Content>**Old seattle getaway ...**

<Date>Jan 6, 2009

<No. Reader>-1

<No. Helpful>-1

<Overall>**5**

<Value>5

<Rooms>5

<Location>5

<Cleanliness>5

<Check in / front desk>5

<Service>5

<Business service>5

PRÉ-PROCESSAMENTO TEXTUAL

- Pontuação – retirada de caracteres especiais e HTML;

```
In [2]: cd Anaconda/  
C:\Users\Alexandre\Anaconda
```

```
In [3]: cd Examples/  
C:\Users\Alexandre\Anaconda\Examples
```

```
In [4]: cd MLiA_SourceCode/  
C:\Users\Alexandre\Anaconda\Examples\MLiA_SourceCode
```

```
In [5]: cd minicurso/  
C:\Users\Alexandre\Anaconda\Examples\MLiA_SourceCode\minicurso
```

```
In [6]: import recupera_texto
```

```
In [7]: recupera_texto.recuperaOpiniaao()
```

STOPWORDS

- *Stopwords* – a, the, an, at, in, on...;
- Lista de stopwords no link:

<http://www.ranks.nl/stopwords>

```
[8]: import remove_stopwords
```

```
[9]: remove_stopwords.retiraStopwords()
```

NEGAÇÃO

- Negação – mudança no sentido de palavras precedidas de no, not ou nothing;
- Exemplo: not clean -> not_clean;

```
[9]: import trata_negacao
```

```
[10]: trata_negacao.trataNegacao()
```

OPINIÃO COM TRATAMENTO TEXTUAL

- “My Wife and I, and some friends, stayed here after this years grand national. We managed to book a double executive room for £80 and thought we were on to a winner - how wrong we were. After attending the national we arrived at the hotel to find out that there was no booking for us”



- “wife some friend stayed here after year grand national managed book double executive room thought were win how wrong were attend national we arrived hotel find out that there was no_book us”

BAG-OF-WORDS

- Unigramas e Bigramas;

-Exemplo: *“Great Hotel lovely staff great location great_hotel hotel_lovely lovely_staff staff_great great_location”*

- Testes com unigramas e unigramas+bigramas (n-gramas).

N-gramas	Exemplos
Unigramas	great, lovely, hotel,
Bigramas	great_hotel, hotel_lovely, lovely_staff
Trigramas	great_hotel_lovely, hotel_lovely_staff

BAG-OF-WORDS

- Características mais comuns utilizadas em análise de texto e tem sido muito efetiva em análise de sentimento [Liu, 2012];
- São relacionados com a frequência de cada *token*.
- Ver o q tem q ser comentado

Unigramas

```
[11]: import criar_unigramas  
[12]: criar_unigramas.criaUnigramas()
```

Unigramas e bigramas

```
[10]: import criar_ngramas  
[11]: criar_ngramas.criaNGramas()
```

SELEÇÃO DE CARACTERÍSTICAS

- Fundamental para a escolha dos n-gramas para o treinamento de algoritmos de aprendizado;
- A seleção de características pode melhorar significativamente o desempenho da classificação;
- Esta etapa consiste na escolha de n-gramas que serão utilizadas como atributos de treinamento.

GANHO DE INFORMAÇÃO

- Ganho de informação: mede o *goodness* de um termo de acordo com a presença ou falta;
- Presença ou falta de um termo t , onde o IG de um termo é dado por:

$$IG(t) = -\sum_{i=1}^Z P(c_i) \log P(c_i) + \frac{P(t) \sum_{i=1}^Z P(c_i|t) \log P(c_i|t)}{P(\bar{t}) \sum_{i=1}^Z P(c_i|\bar{t}) \log P(c_i|\bar{t})}$$

EXEMPLO

Classe	Termo	
	good	\overline{good}
Sim	10	2
Não	1	5

- $Ent(classe) = -((2/3 * \log(2/3) + (1/3 * \log(1/3))) = 0,919$
- $Ent(classe | good) = -((5/9 * \log(5/9) + (1/18 * \log(1/18))) = 0,224$
- $Ent(classe | \overline{good}) = -((1/9 * \log(1/9) + (5/18 * \log(5/18))) = 0,020$
- $Info(good) = - Ent(classe) + Ent(classe | \overline{good}) + Ent(classe | good) = (0.919 - 0.224 - 0.02) = \mathbf{0.675}$

GANHO DE INFORMAÇÃO - PYTHON

- Unigrama

```
[15]: import calcula_ig_unigrama
```

```
[16]: calcula_ig_unigrama.calculaIg(250)
```

- Unigramas e Bigramas

```
[14]: import calcula_ig_ngrama
```

```
[15]: calcula_ig_ngrama.calculaIg(250)
```

GANHO MÉDIO DE INFORMAÇÃO

- Ganho médio de informação: normaliza a contribuição de uma característica;
- *Split Information*: são calculados por meio da informação obtida pela divisão de um documento de treinamento P em v partes, na qual v corresponde ao número de atributos

$$SplitInfo(t) = -\sum_{j=1}^v \frac{|P_j|}{|P|} \log \frac{|P_j|}{|P|}$$

- Por fim, o ganho médio é dado por:

$$Gain\ Ratio(t) = Information\ Gain(t)/SplitInfo(t).$$

EXEMPLO

Classe	Termo	
	<i>good</i>	\overline{good}
Sim	10	2
Não	1	5

- $Ent(classe) = -((2/3 * \log(2/3) + (1/3 * \log(1/3))) = 0.919$
- $Ent(classe | good) = -((5/9 * \log(5/9) + (1/18 * \log(1/18))) = 0.224$
- $Ent(classe | \overline{good}) = -((1/9 * \log(1/9) + (5/18 * \log(5/18))) = 0.020$
- $Info(good) = - Ent(classe) + Ent(classe | \overline{good}) + Ent(classe | good) = (0.919 - 0.224 - 0.02) = \mathbf{0.675}$
- $SplitInfo = -((5/9 * \log(5/9) + (1/18 * \log(1/18))) = 0.224$
- $Gain(good) = 0.675/0.224 = \mathbf{3,013}$

GANHO MÉDIO DE INFORMAÇÃO - PYTHON

- Unigrama

```
[17]: import calcula_gr_unigrama
```

```
[18]: calcula_gr_unigrama.calculaGr(250)
```

- Unigramas e Bigramas

```
[16]: import calcula_gr_ngrama
```

```
[17]: calcula_gr_ngrama.calculaGr(250)
```

CHI QUADRADO

- Chi-quadrado: representa a associação entre uma característica e a classe correspondente;
- Vetores criados a partir de *tokens* mais comuns por meio de unigramas ou n-gramas;
- Fórmula:

$$\text{CHI}(t, c_i) = \frac{N \cdot (AD - BE)^2}{(A+E) \cdot (B+D) \cdot (A+B) \cdot (E+D)} \text{ and } \text{CHI}_{\max} = \max_i(\text{CHI}(t, c_i))$$

- **A** é o número de vezes que t e c_i ocorrem simultaneamente;
- **B** é o número de vezes que t ocorre sem c_i ;
- **E** é o número de vezes que c_i ocorre sem t ;
- **D** é o número de vezes que nem c_i nem t ocorrem, e;
- **N** é o total de documentos

CHI QUADRADO

Classe	Termo	
	<i>good</i>	\overline{good}
Sim	10	2
Não	1	5
Neutro	3	3

- A = 10
- B = 4
- E = 2
- D = 8
- N = 24

$$CHI(good, sim) = \frac{24 * (10.8 - 4.2)^2}{(10+2) * (4+8) * (10+4) * (2+8)} = 6,17$$

- A = 1
- B = 13
- E = 5
- D = 5
- N = 24

$$CHI(good, nao) = \frac{24 * (1.5 - 13.5)^2}{(1+5) * (13+5) * (1+13) * (5+5)} = 5,71$$

- A = 3
- B = 11
- E = 3
- D = 7
- N = 24

$$CHI(good, neutro) = \frac{24 * (3.7 - 11.3)^2}{(3+3) * (11+7) * (3+11) * (3+7)} = 0,23$$

CHI QUADRADO - PYTHON

- Unigrama

```
[13]: import calcula_chi_unigrama
```

```
[14]: calcula_chi_unigrama.calculaChi(250)
```

- Unigramas e Bigramas

```
[12]: import calcula_chi_ngrama
```

```
[13]: calcula_chi_ngrama.calculaChi(250)
```

VETORIZAÇÃO

- Etapa que tem como objetivo transformar uma frase em um vetor de características, onde os atributos correspondem aos n-gramas selecionados.
- Estes atributos são configurados de acordo com frequência dos mesmos em relação a uma opinião;
- Frequência ou *tfidf*.

FREQUÊNCIA

- Definição:

Opinião: *Great Hotel lovely staff great location stayed 2 nights hen party hotel close all bars night clubs shopping would definitely stay again **great_hotel** hotel_lovely lovely_staff staff_great great_location location_stayed 2_nights stay_again. Rating: 5.*

Words (15): great, lovely, worst, location, stayed, close, **great_hotel**, terrible, clean, hotel_lovely, nice, comfortable, handy, stay_again, good.

Word	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Rating
Freq.	2	1	0	1	1	1	1	1	0	1	0	0	0	1	0	5

FREQUÊNCIA

- Binário

```
[20]: import prepara_vetor_frequencia_bin
```

```
[21]: prepara_vetor_frequencia_bin.Vectorizer(250)
```

- 5 classes (*ratings*)

```
[18]: import prepara_vetor_frequencia
```

```
[19]: prepara_vetor_frequencia.Vectorizer(250)
```

TFIDF

- Definição

Opinião: *Great Hotel lovely staff great location stayed 2 nights hen party hotel close all bars night clubs shopping would definitely stay again great_hotel hotel_lovely lovely_staff staff_great great_location location_stayed 2_nights stay_again. Rating: 5.*

Words (15): great, lovely, worst, location, stayed, close, great_hotel, terrible, clean, hotel_lovely, nice, comfortable, handy, stay_again, good.

$$w_i = tf_i \cdot \log \frac{N}{df_i}$$

Word	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Rating
w_i	0.8	0.5	0	0.5	0.5	0.5	0.5	0.5	0	0.5	0	0	0	0.5	0	5

TFIDF

- Binário

```
[21]: import prepara_vetor_tfidf_bin
```

```
[22]: prepara_vetor_tfidf_bin.Vectorizer(250)
```

- 5 classes (*ratings*)

```
[19]: import prepara_vetor_tfidf
```

```
[20]: prepara_vetor_tfidf.Vectorizer(250)
```

A FERRAMENTA WEKA

- Weka é um pacote desenvolvido pela Universidade de Waikato, em 1993;
- Agrega algoritmos para mineração de dados na área de Inteligência Artificial;
- Possui uma série de heurísticas para mineração de dados relacionadas à classificação, regressão, clusterização, regras de associação.

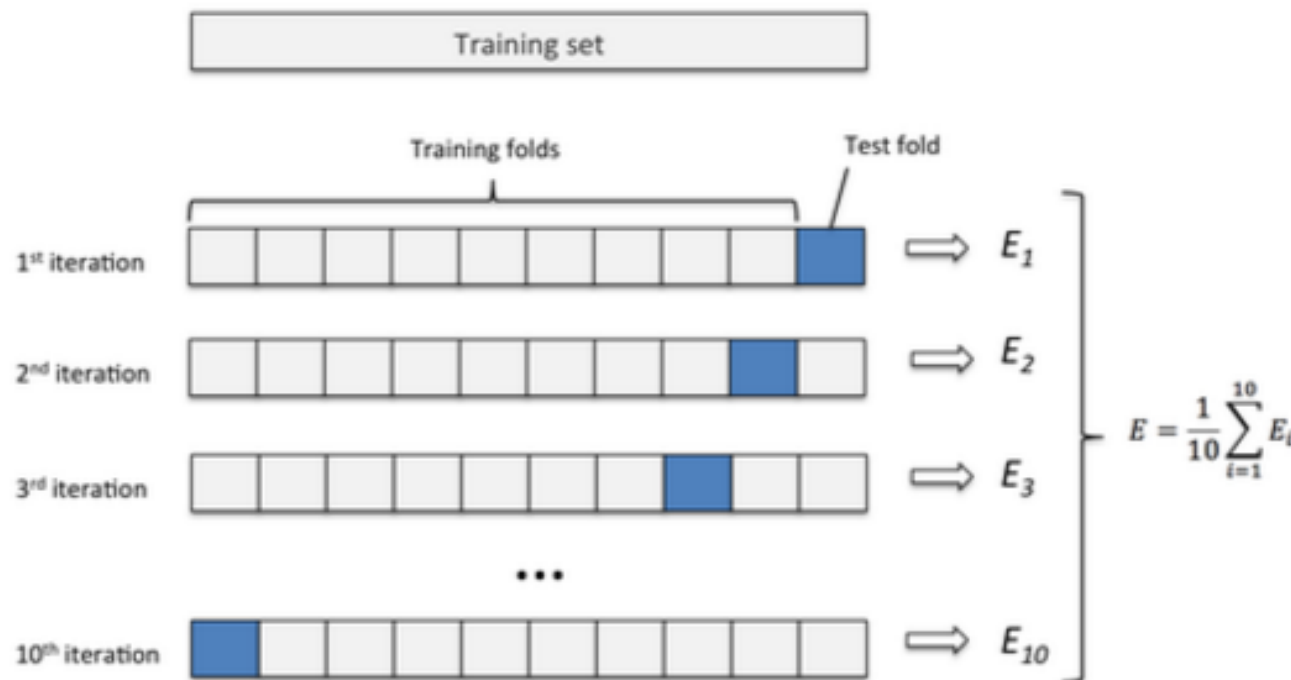


MODELOS DE CLASSIFICAÇÃO

- Naive Bayes
 - Naive Bayes;
 - NaiveBayes Multinomial.
- SVM
 - SMO;
 - LibSVM.
- Árvores de Decisão
 - J48.
- kNN
 - Ibk.
- Modelos adaptados
 - One-vs-all;
 - One-vs-one.

10-FOLD CROSS VALIDATION

- A validação cruzada estima como o modelo construído irá se comportar em novos dados.
- O *k-fold cross validation* consiste em dividir a base em k pedaços. Para cada pedaço, estimamos o método sem a presença desta parte e verificamos o erro médio no pedaço não utilizado durante o treino.



MEDIDAS AVALIATIVAS

- A acurácia é calculada como:

$$A = \frac{\text{número de exemplos classificados corretamente}}{\text{total de exemplos}}$$

- A precisão é dada por:

$$P = \frac{\text{número de corretas predições positivas}}{\text{número de predições positivas}}$$

- O *recall* é dado pela seguinte fórmula:

$$R = \frac{\text{número de corretas predições positivas}}{\text{número de exemplos positivos}}$$

EXEMPLO

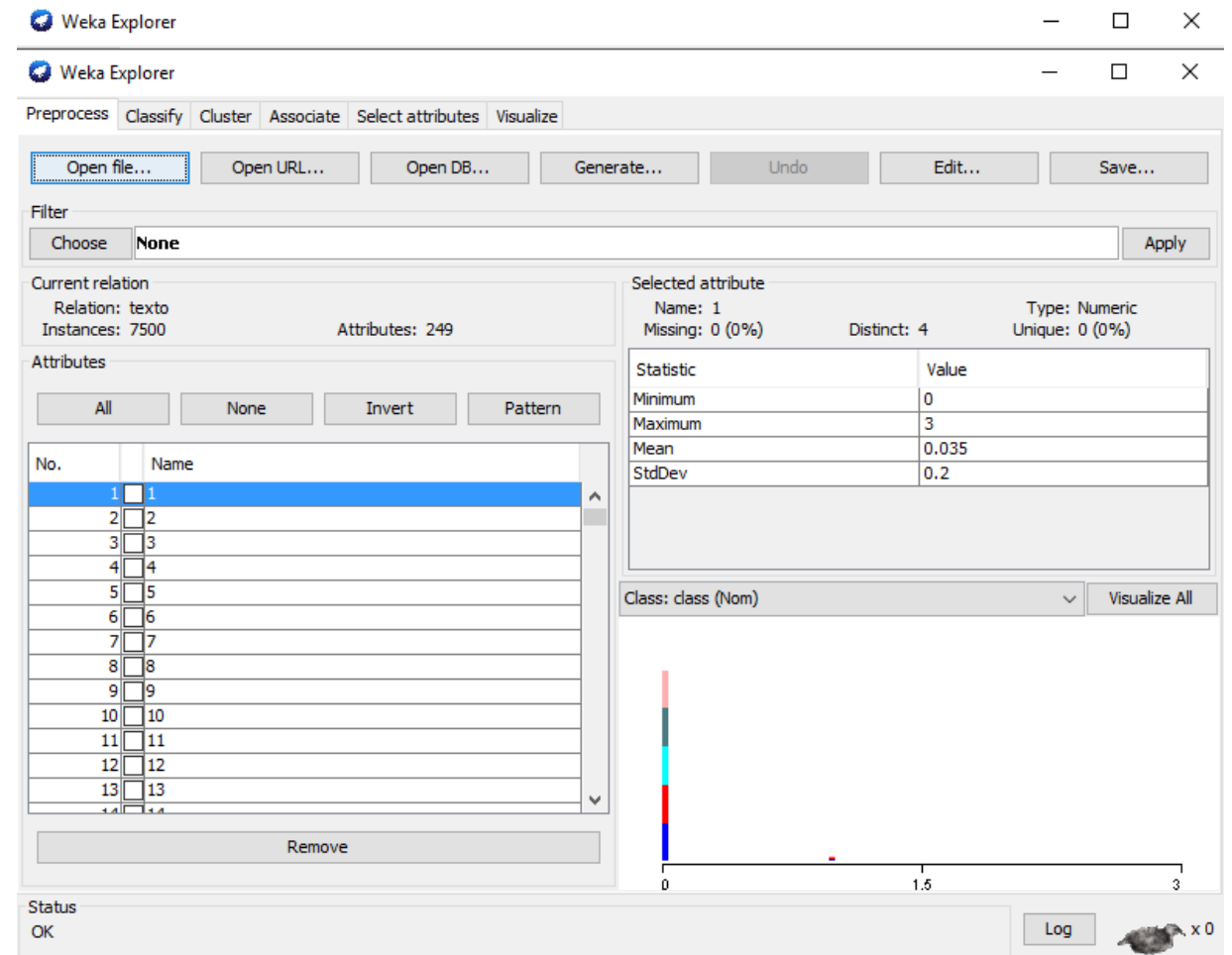
a	b	c	d	e	Total	← classificado como
1140	276	61	10	13	1500	a=1
497	502	380	96	25	1500	b=2
132	260	773	281	54	1500	c=3
47	97	228	648	480	1500	d=4
19	36	45	267	1133	1500	e=5
1835	1171	1487	1302	1705	7500	

$$A = \frac{\text{número de exemplos classificados corretamente}}{\text{total de exemplos}} = (1140+502+773+648+1133)/7500 = 0,5594$$

$$P = \frac{\text{número de corretas predições positivas}}{\text{número de predições positivas}} = 1140/1835 = 0,6212$$

$$R = \frac{\text{número de corretas predições positivas}}{\text{número de exemplos positivos}} = 1140/1500 = 0,76$$

IMPORTANDO ARQUIVOS NO WEKA



NAIVE BAYES

- Uma variação da teoria de decisão Bayesiana Fórmula

$$P(c | d) = \frac{P(c)P(d | c)}{P(d)}$$

- O modelo multinomial captura a frequência de uma palavra no conjunto de opiniões. Para associar a um novo exemplo t uma classe c_i , a classe com maior probabilidade $c^* = \operatorname{argmax} P(c_i | t)$ é considerada. Na equação abaixo é mostrado como o cálculo das probabilidades para cada classe $c_i \in c$ é realizado.

$$P_{NB}(c_i | t) = P(c_i) (\prod_{j=1}^D P(t_j | c_i)),$$

no qual t é um termo, i é o número da classe e D é o conjunto de opiniões.

NAIVE BAYES

	Num	Palavras	Classe
Treinamento	1	Chinese Beijing Chinese	1
	2	Chinese Chinese Shangai	1
	3	Chinese Macao	1
	4	Tokyo Japan Chinese	2
Teste	5	Chinese Chinese Chinese Tokyo Japan	?

Probabilidades:

$$P(\text{Chinese} | 1) = (5+1)/(8+6) = \frac{3}{7}$$

$$P(\text{Tokyo} | 1) = (0+1)/(8+6) = \frac{1}{14}$$

$$P(\text{Japan} | 1) = (0+1)/(8+6) = \frac{1}{14}$$

$$P(\text{Chinese} | 2) = (1+1)/(3+6) = \frac{2}{9}$$

$$P(\text{Tokyo} | 2) = (1+1)/(3+6) = \frac{2}{9}$$

$$P(\text{Japan} | 2) = (1+1)/(3+6) = \frac{2}{9}$$

Classes:

$$P(1) = \frac{3}{4}$$

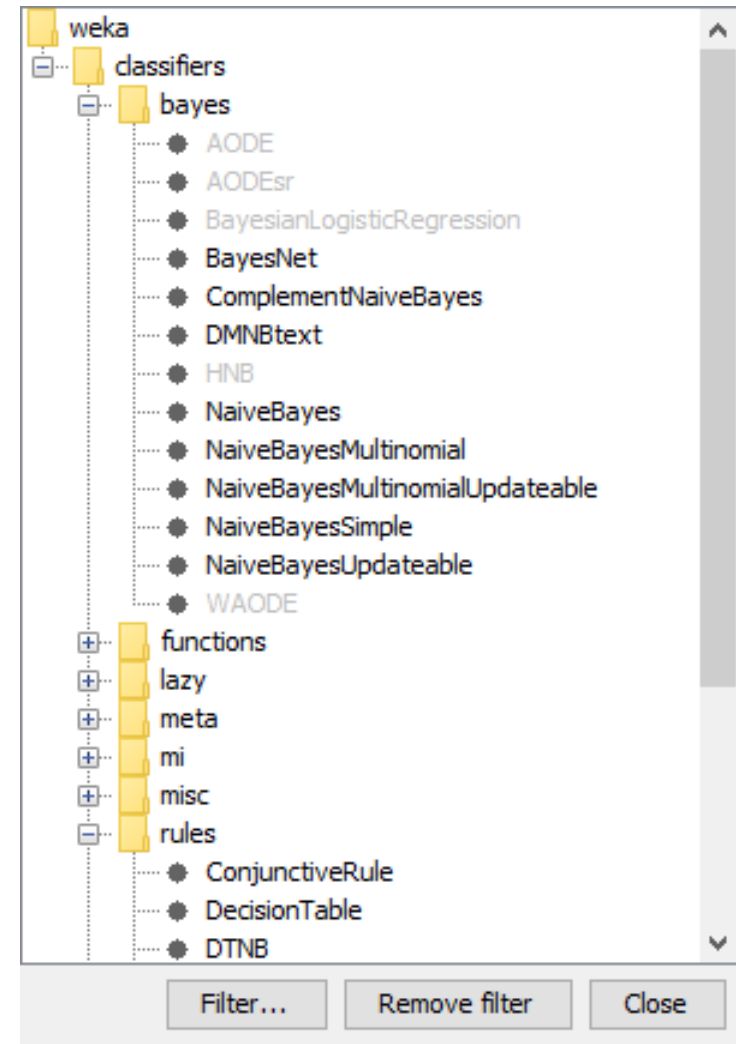
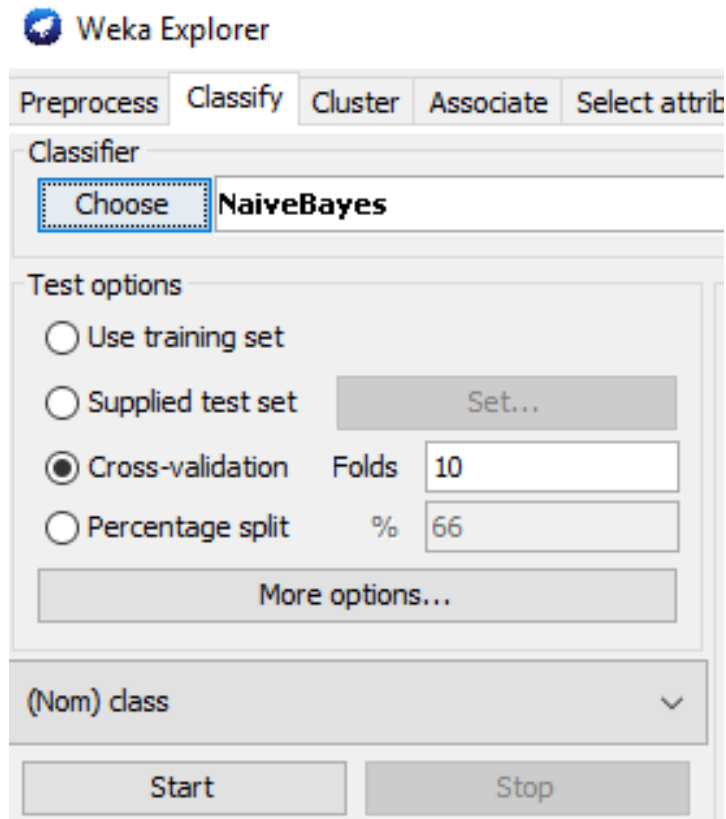
$$P(2) = \frac{1}{4}$$

Classe Final:

$$P(1 | d5) = 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0,0003$$

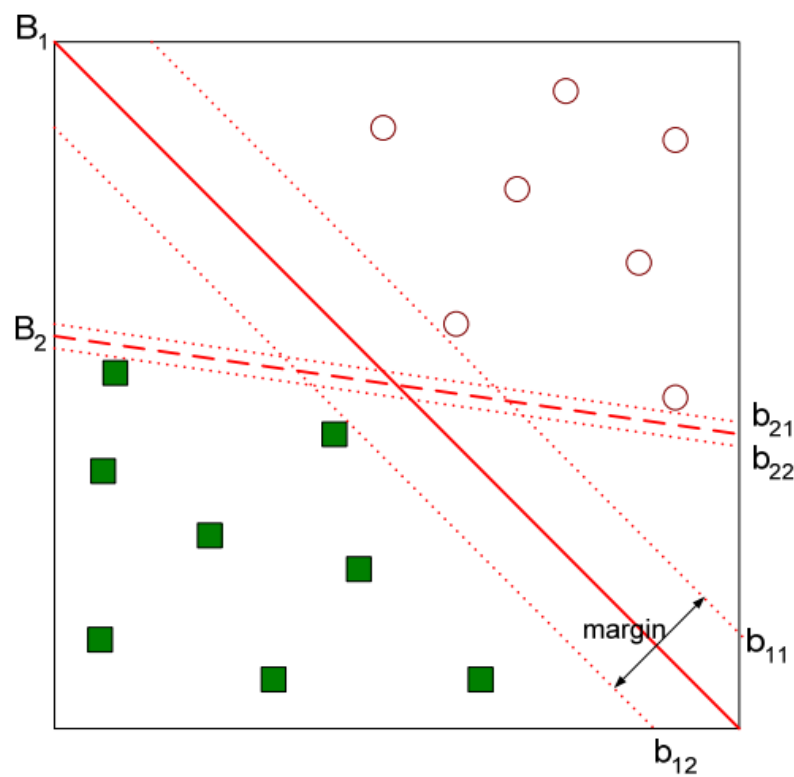
$$P(2 | d5) = 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$$

NAIVE BAYES



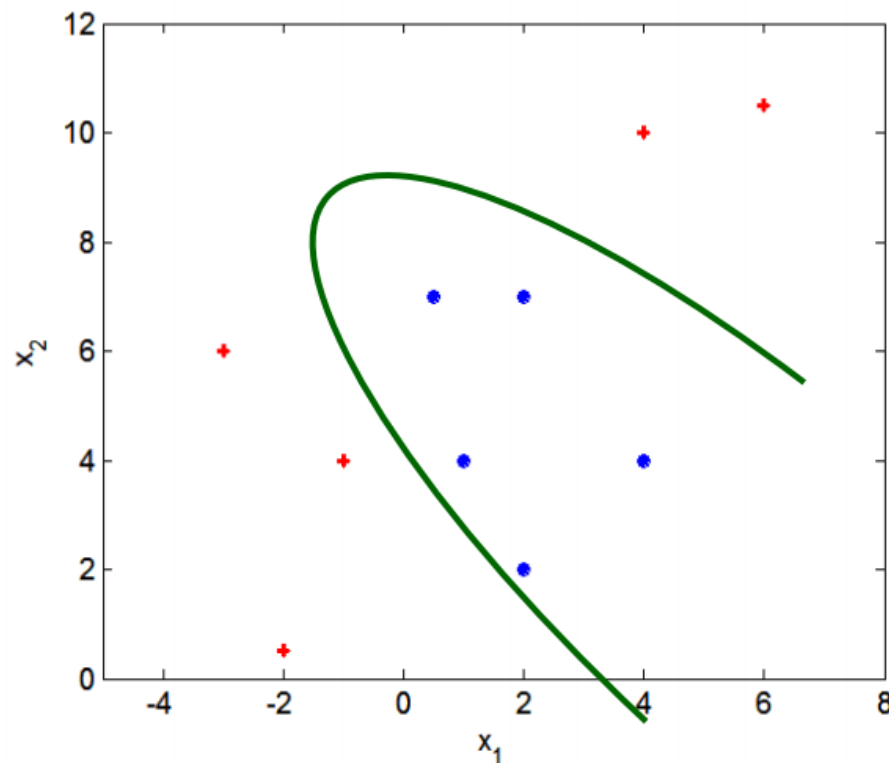
SVM

- Em um conjunto de treinamento, existem infinitas linhas separando duas classes;
- A ideia principal do modelo de máquina de vetores de suporte é encontrar as margens ótimas em relação a um hiperplano separador h .

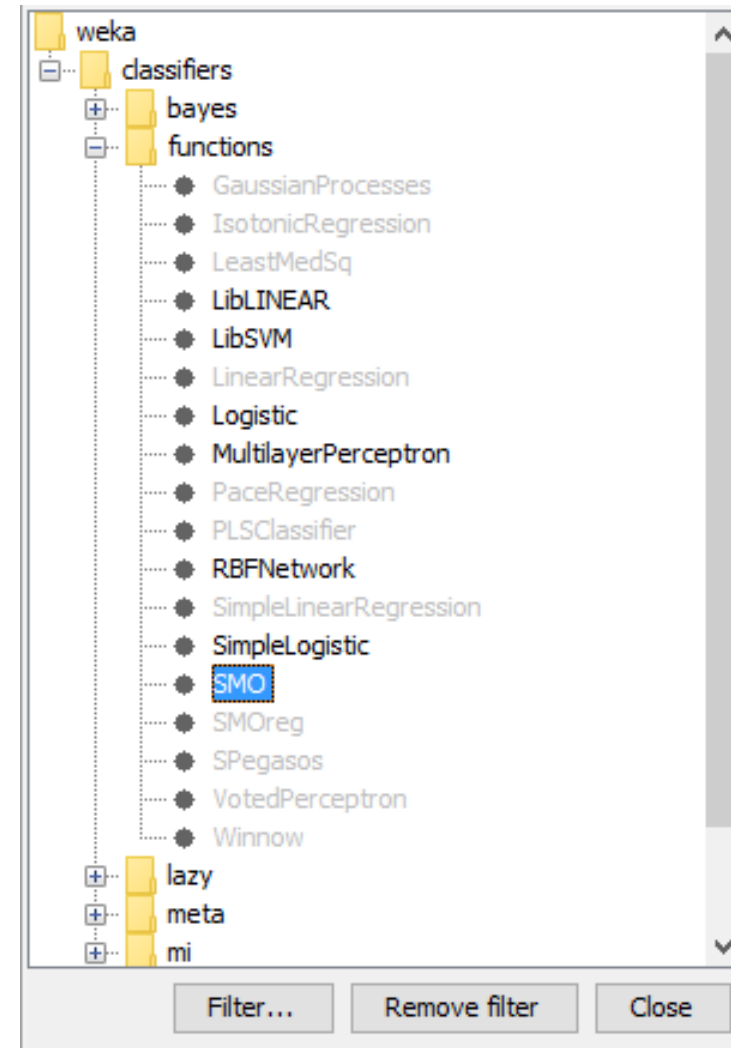
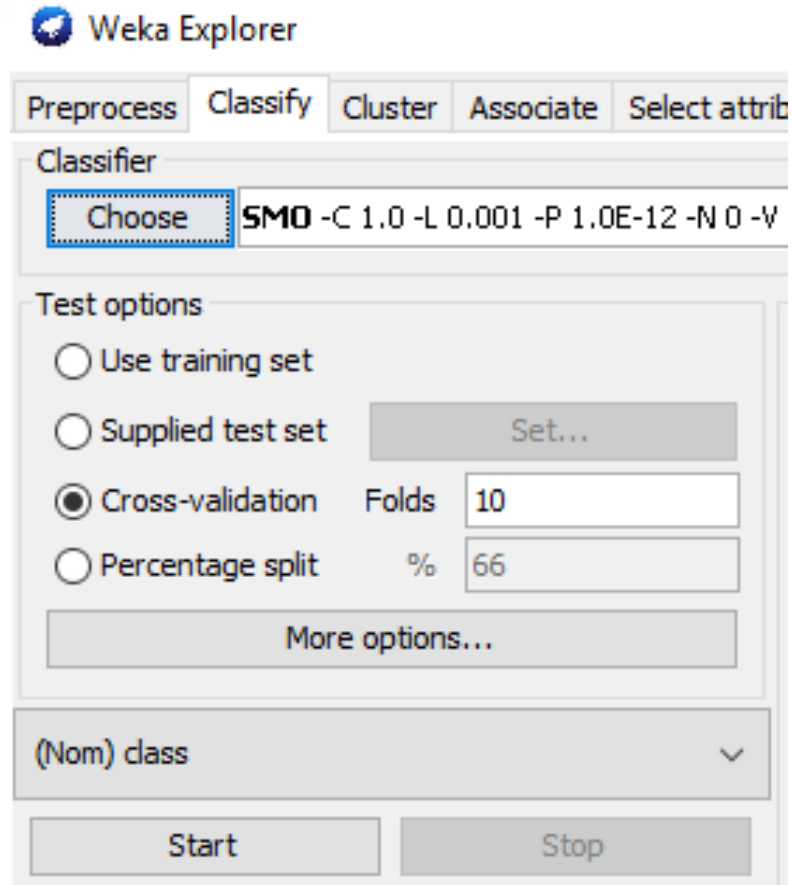


SVM

- Os pontos mais próximos são chamados de vetores de suporte;
- Considerado por [Joachims, 98], como o melhor classificador de textos;
- Não-linear.



SVM

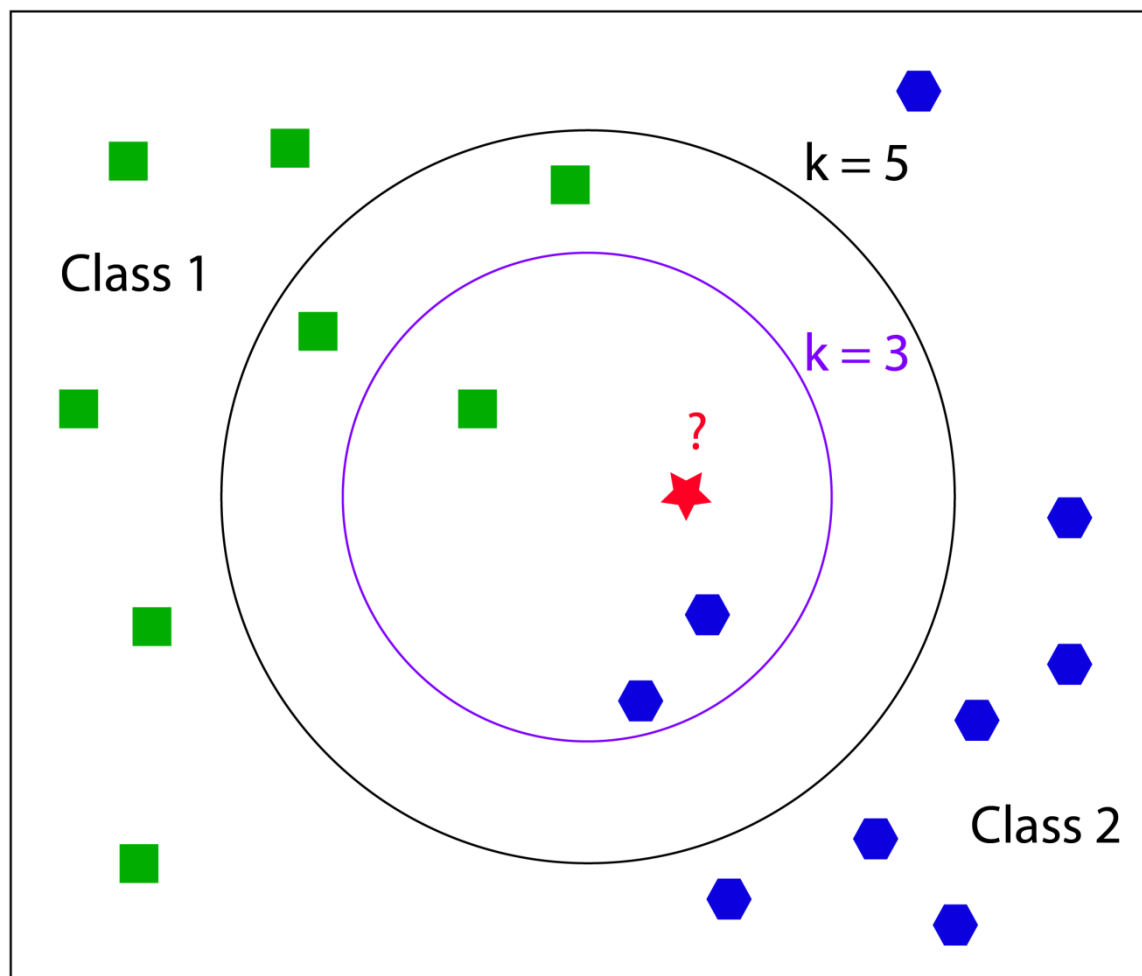


KNN DE DECISÃO

- Os k-vizinhos mais Próximos ou k-Nearest Neighbors (kNN) é um método baseado em instâncias que aprende com o simples armazenamento dos dados de treinamento.
- A partir dos k vizinhos mais parecidos, ele escolhe o dado com os k mais similares com o que será classificado e atribui uma nova classe a ele.
- Exemplo: distância Euclidiana

$$u = \sqrt{(xA_0 - xB_0)^2 + (xA_1 - xB_1)^2}$$

KNN



KNN

Choose **IBk** -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A {\weka.core.EuclideanDistance -R first-last\}"

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds

☐ Percentage split %

More options...

(Nom) class ▼

Start Stop

Result list (right-click for options)

weka.gui.GenericObjectEditor

weka.classifiers.lazy.IBk

About

K-nearest neighbours classifier. More Capabilities

KNN

crossValidate ▼

debug ▼

distanceWeighting ▼

meanSquared ▼

nearestNeighbourSearchAlgorithm Choose **LinearNNSearch** -A "weka.core.EuclideanDistance -R first-last\"

windowSize

Open... Save... OK Cancel

weka

- classifiers
 - bayes
 - functions
 - lazy
 - IB1
 - IBk**
 - KStar
 - LBR
 - LWL
 - meta
 - mi
 - misc
 - rules
 - trees

Filter... Remove filter Close

ÁRVORES DE DECISÃO

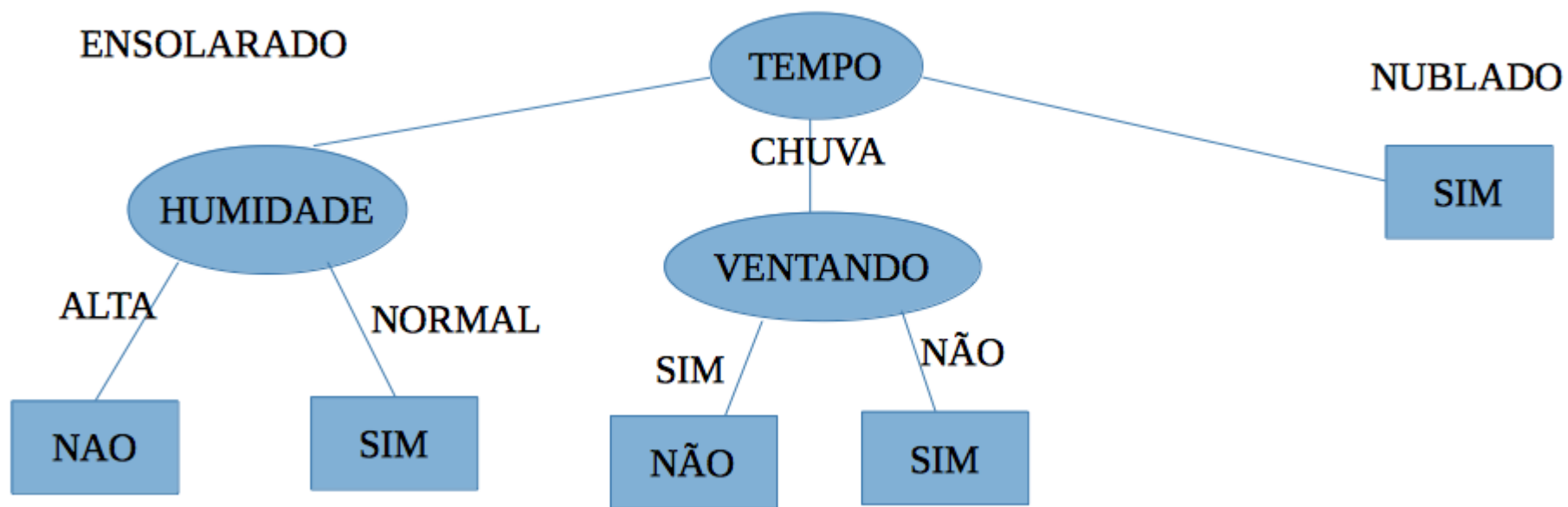
- As árvores de decisão consistem em um método de aproximação discreta do alvo, na qual a função de aprendizado é representada por uma árvore de decisão.
- Para escolher um atributo que divida a árvore, o objetivo é escolher o atributo que possui o maior ganho (*gain*). Esse ganho é definido por meio da redução da Entropia:

$$\text{Entropia}(P) = (-(P_+ \log_2 P_+) - (P_- \log_2 P_-))$$

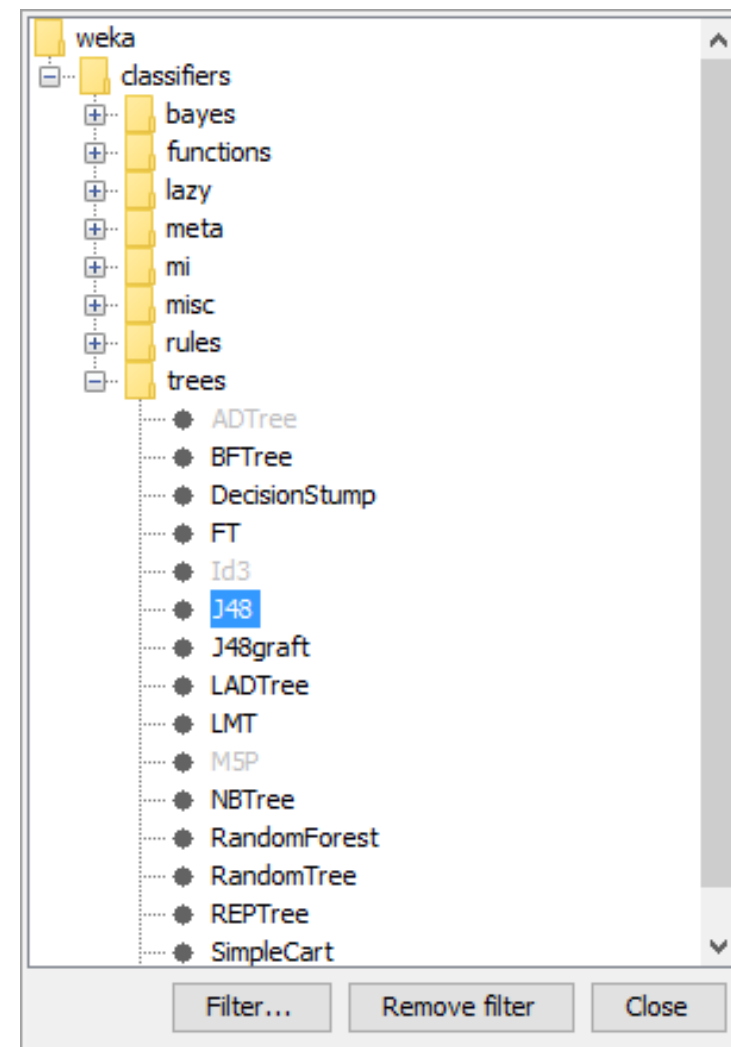
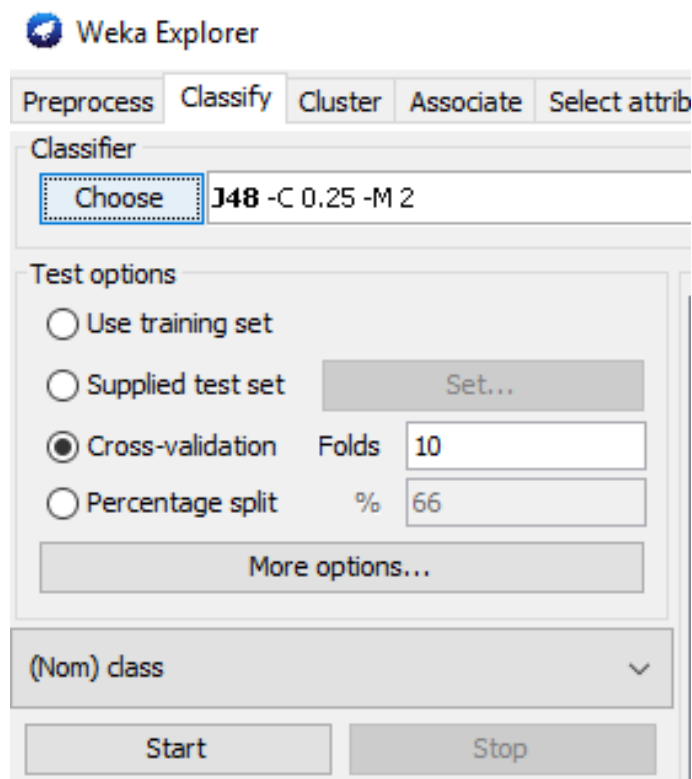
- Com a entropia definida, o ganho é dado por:

$$\text{Ganho}(P,t) = \text{Entropia}(P) - \sum_j^{\text{valores}(n)} \frac{P_j}{P} \text{Entropia}(P_j)$$

ÁRVORES DE DECISÃO



ÁRVORES DE DECISÃO



MODELOS ADAPTADOS – ONE-VS-ONE

- No modelo OvO, cada classe c_i é comparada com outra classe c_k , onde $k, i = 1..n$ e $i \neq k$, dado que n é o número de classes.
- O número de etapas para a classificação é dado por $\frac{n(n-1)}{2}$.
- Na fase de classificação, a classe escolhida é baseada em uma votação direta dada pelo maior valor de acordo com

$$(x) = \arg \max_i \left(\sum_j f_{ij}(x) \right).$$

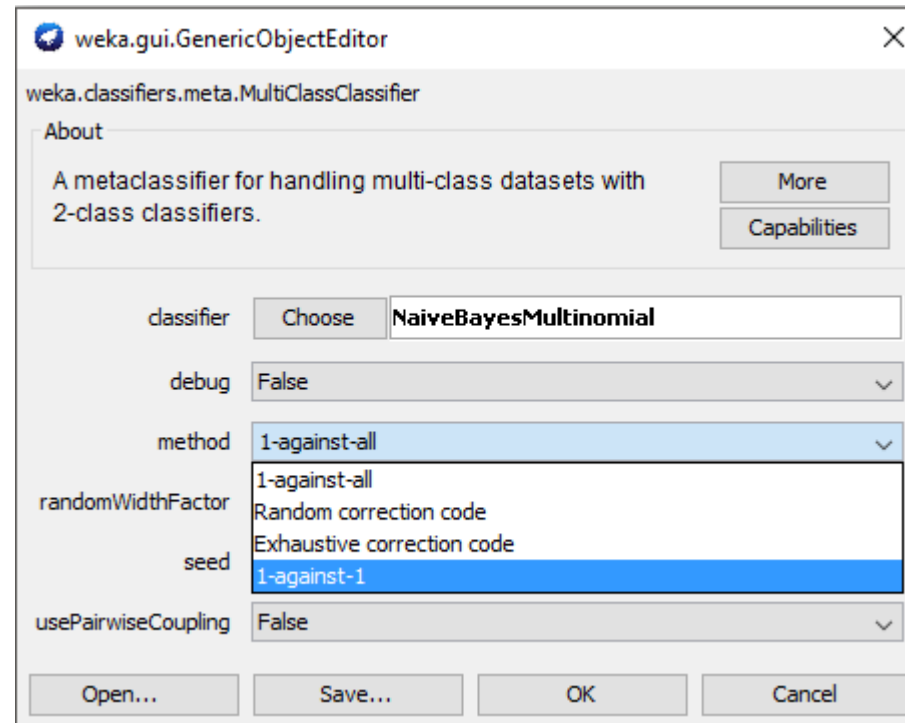
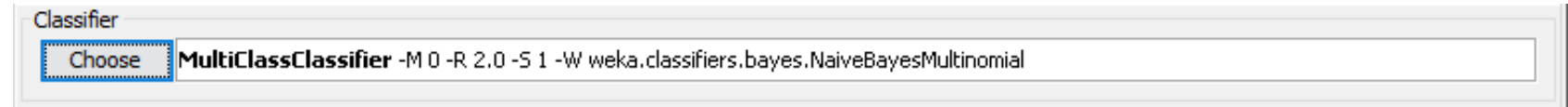
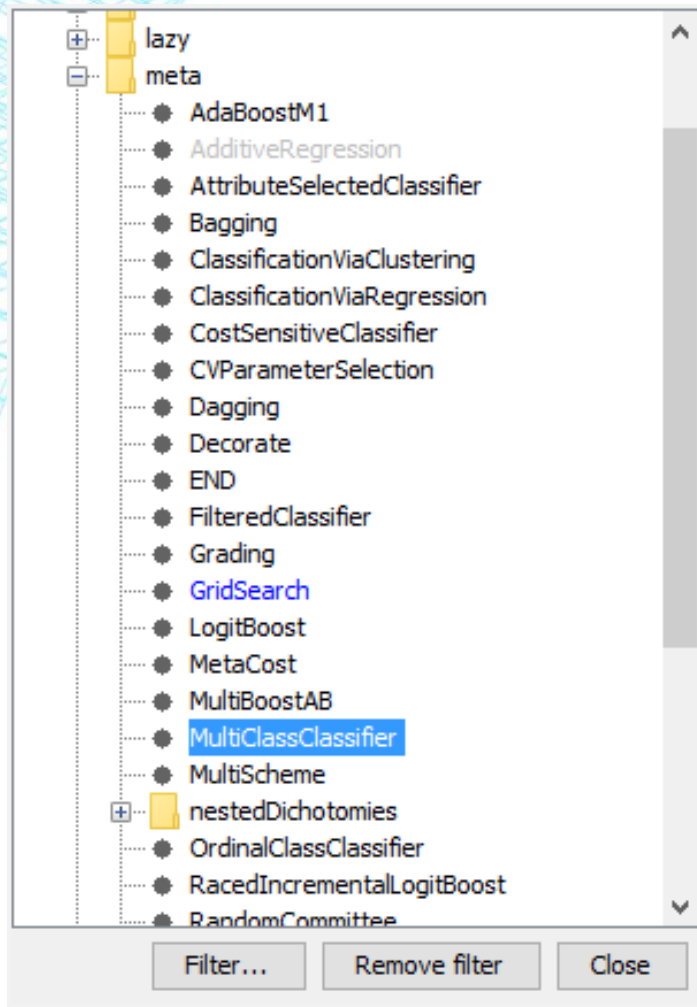
EXEMPLO

- Para um problema com 4 classes {1, 2, 3, 4}, o OvO cria 6 classificadores (1-2, 1-3, 1-4, 2-3, 2-4, 3-4);
- Exemplo de predição para uma nova instância a .

Classificador	$f(a)=$
1-2	2
1-3	1
1-4	1
2-3	2
2-4	2
3-4	3

	Votos para cada classe			
Classe	1	2	3	4
Número de votos	2	3	1	0

MODELOS ADAPTADOS – ONE-VS-ONE



MODELOS ADAPTADOS – ONE-VS-ALL

- Em um modelo OvA n classificadores são construídos, isto é, para cada comparação entre uma classe e as demais, um classificador é construído.
- No processo de classificação, dada uma nova instância a , a predição é dada por:

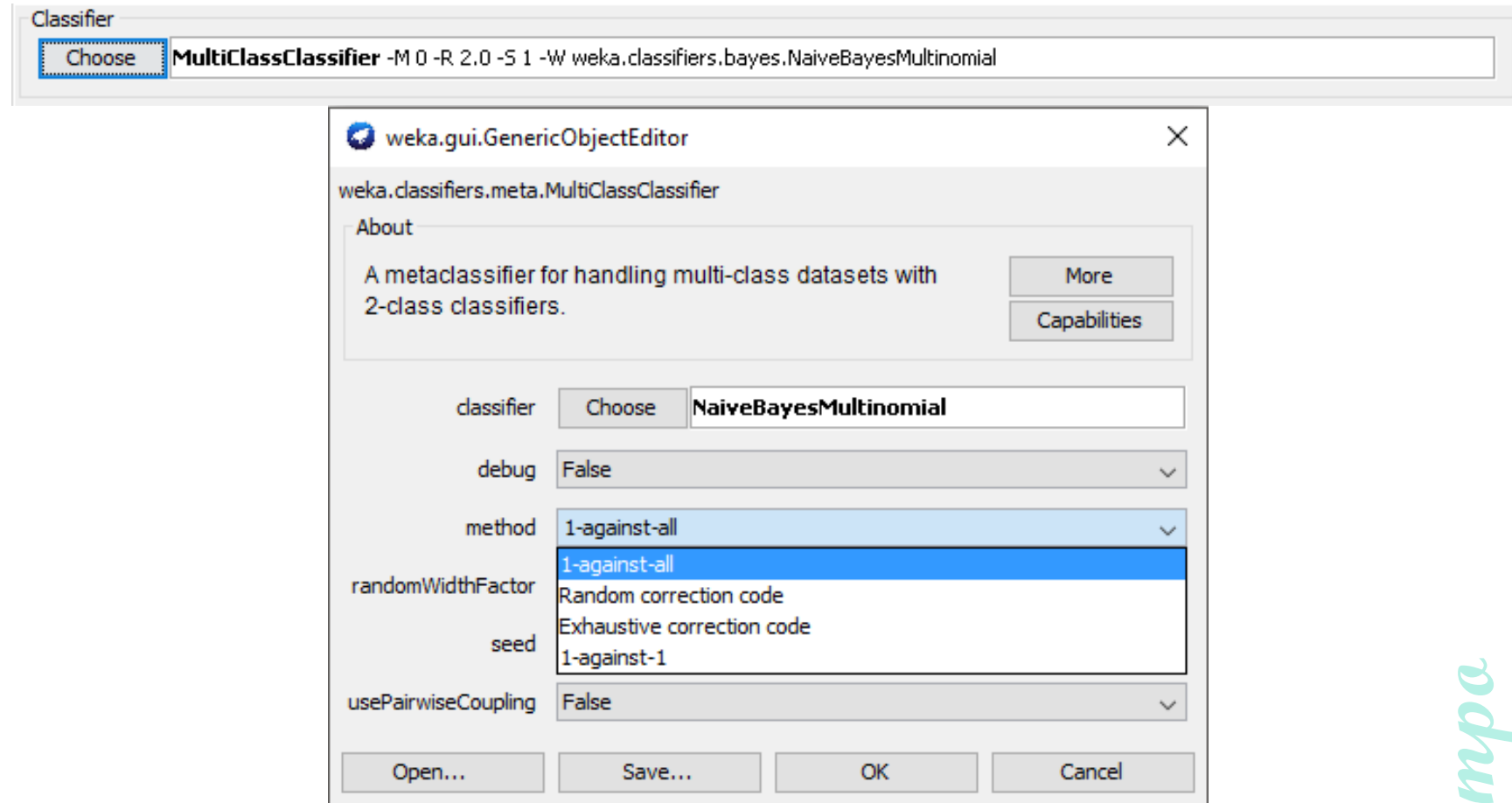
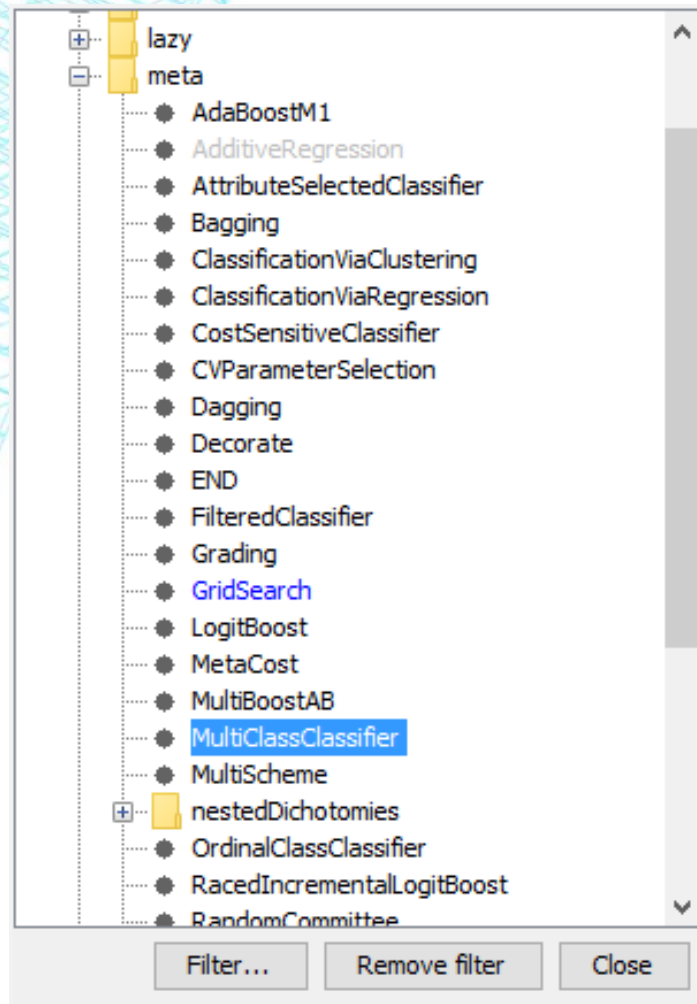
$$f(x) = \arg \max_i f_i(x)$$

EXEMPLO

- Avaliando o modelo OvA para o mesmo número de classes, 4 classificadores são criados (1 vs {234}, 2 vs {134}, 3 vs {124}, 4 vs {123});
- Para uma nova instância a :

f(a)		Votos			
Classificador		1	2	3	4
1 vs {234}	Outra	0	0,333	0,333	0,333
2 vs {134}	2	0	1	0	0
3 vs {124}	Outra	0,333	0,333	0	0,333
4 vs {123}	Outra	0,333	0,333	0,333	0
Total		0,666	1,999	0,666	0,666

MODELOS ADAPTADOS – ONE-VS-ALL



ANÁLISE DOS RESULTADOS

- Dependência do domínio.
- Melhores técnicas?
- Melhores algoritmos?

ANÁLISE DE SENTIMENTOS - BINÁRIO

Autores	Domínio	Features	Algoritmos	Acurácia (best) %
Pang et al. 2002	Revisões de filmes	POS, unigramas, bigramas, position, adjectives	NB, MaxEnt and SVM	82.9 (SVM + unigramas)
Mullen and Collier 2004	Revisões de filmes e discos	Unigramas, Lemmas, Osgood and Turney	SVM	Filmes – 86 (SVM + Turney and Lemmas) Discos – 89 (SVM + PMI/Osgood + Lemmas)
Matsumoto et al. 2005	Revisão de filmes	unigramas, bigramas, frequentes subsequências de palavras e sub-árvores dependentes	SVM	93.7 (SVM + unigramas + bigramas, frequentes subsequências de palavras)
Tan and Zhang 2008	Opiniões sobre educação, filmes e casa	MI, IG, DF and Chi	Classificador centroide, kNN, NB, Winnow e o SVM	90.6 (SVM + IG)
Go et al. 2009	tweets	Sentiment words, bigramas and unigramas	NB, MaxEnt and SVM	83.0 (MaxEnt with unigram + bigram)
Paltoglou and Thelwall 2010	Revisão de filmes	Unigramas, document frequency and term frequency	SVM	96.9 (SVM + BM25 tf + BM25 delta idf variant) ^b
Xia et al., 2011	Opiniões sobre livros, eletrônicos, DVD's e artigos de cozinha	POS and word-relation (WR)	NB, SVM and MaxEnt	86.85 - Movie (MaxEnt + POS) 88.65 – Kitchen (NB + WR)
Sharma and Dey 2012	Revisão de filmes	IG, Mi, GR, Chi and Belief-F	NB, SVM, MaxEnt, DT, kNN, Adaboost and Winnow	90.9 (NB + GR)

APLICAÇÕES DA ANÁLISE MULTICLASSE

Autores	Termos	TEC/AAM*	Classes	Domínio	Acurácia (%)
Pang e Lee, 2005	$N \geq 20$, 50% ou mais em uma classe	Ova+PSP	4	Filmes	54,6
Goldberg e Zhu, 2006	1000 e 5000	SVM Regressão + vetor de palavras	4	Filmes	54,9
Long et al., 2010**	-	Complexidade Kolmogorov e Naive Bayes	5	Hotéis	73,1 – 57,3
Albornoz et al., 2011	114, 330, 353	Vetor de intensidade das características + Logistic	5	Hotéis	46,9
Paltoglou e Thelwall, 2013	-	Arousal + SVM OVA	5	Notícias	51,8

CONCLUSÃO

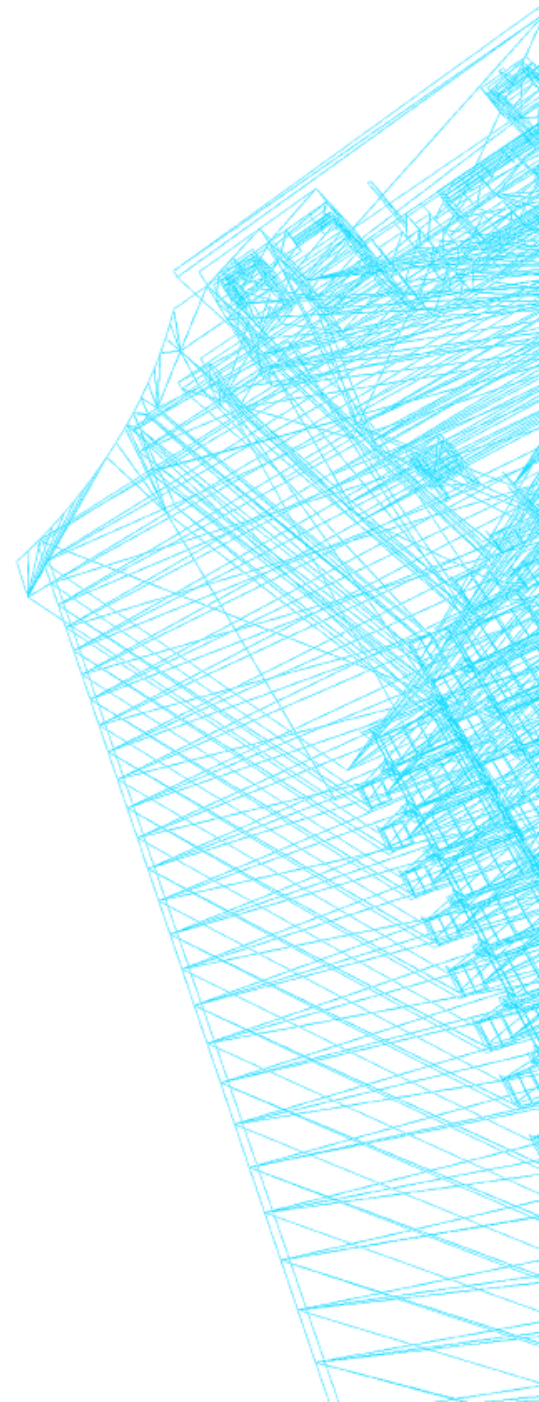
- Técnicas selecionadas de acordo com a importância e a relevância dos trabalhos;
- Modelos podem ser utilizados em sistemas e-commerce para capturar e compreender o sentimento dos usuários;
- Sistemas semelhantes ao sistema *sentiment140*;
- Resultados dos experimentos:

<http://www2.ic.uff.br/PosGraduacao/Dissertacoes/722.pdf>

REFERÊNCIAS

- CAMBRIA, E. et al. **New Avenues in Opinion Mining and Sentiment Analysis**. IEEE Intelligent Systems, n. April, p. 15–21, 2013.
- FRANK, E.; KRAMER, S. **Ensembles of balanced nested dichotomies for multi-class problems**. Lecture Notes in Computer Science, v. 3721 LNAI, p. 84–95, 2004.
- LIU, B. **Sentiment Analysis and Opinion Mining**. Morgan and Claypool Publishers, n. May, 2012
- DE ALBORNOZ, J. C.; PLAZA, L.; GERVÁS, P.; DÍAZ, A. **A joint model of feature mining and sentiment analysis for product review rating**. Advances in information retrieval, p. 55–66, 2011
- PANG, B.; LEE, L. Seeing stars: **Exploiting class relationships for sentiment categorization with respect to rating scales**. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (pp. 115-124). Association for Computational Linguistics. v. 3, n. 1, 2005
- PANG, B.; LEE, L.; VAITHYANATHAN, S. **Thumbs up? Sentiment Classification using Machine Learning Techniques**. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, n. July, p. 79–86, 2002

OBRIQADO





CONTATOS

- E-mail: alexandre.lunardi2@gmail.com