

TRANSFORMAÇÃO S NO DOMÍNIO DOS ATRIBUTOS

Profa. Dra. Marcela Xavier Ribeiro

1

CONTEÚDO

1. CATEGÓRICO PARA INTEIRO
2. CATEGÓRICO PARA BINÁRIO
3. CATEGÓRICO PARA REAL
4. NUMÉRICOS PARA CATEGÓRICOS
5. DISCRETIZAÇÃO
6. QUANTIZAÇÃO

**DE CATEGÓRICO
PARA ...**

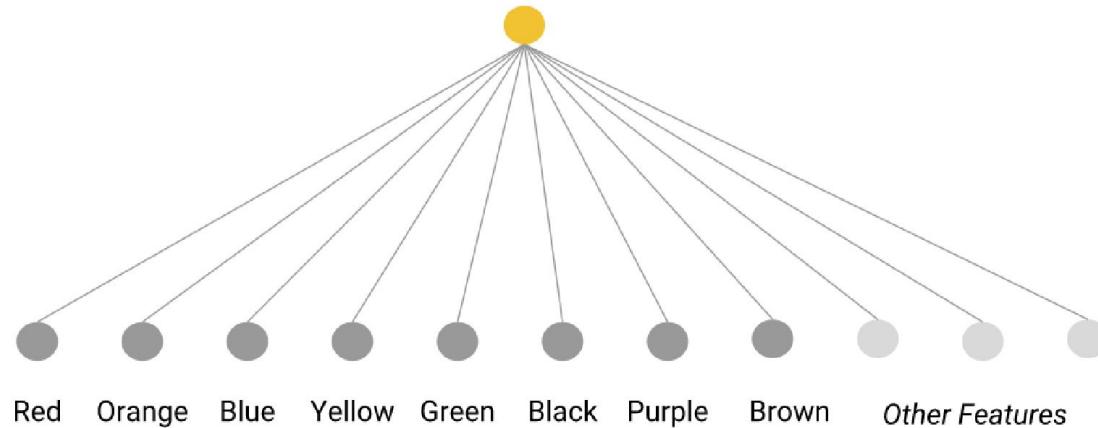
CATEGÓRICOS PARA INTEIROS

- Ordinal Encoder
 - Cada valor categórico é mapeado para um inteiro, de 1 a L (onde L é o número de diferentes valores categóricos).
 - Pode-se usar a ordem alfabética ou outra ordem pré-definida
- Count Encoder
 - Cada valor categórico é mapeado para o número de ocorrências que o mesmo aparece na base de dados
- Hash Encoder
 - Cada valor categórico é mapeado por uma função hash.
- One-hot-Encoder

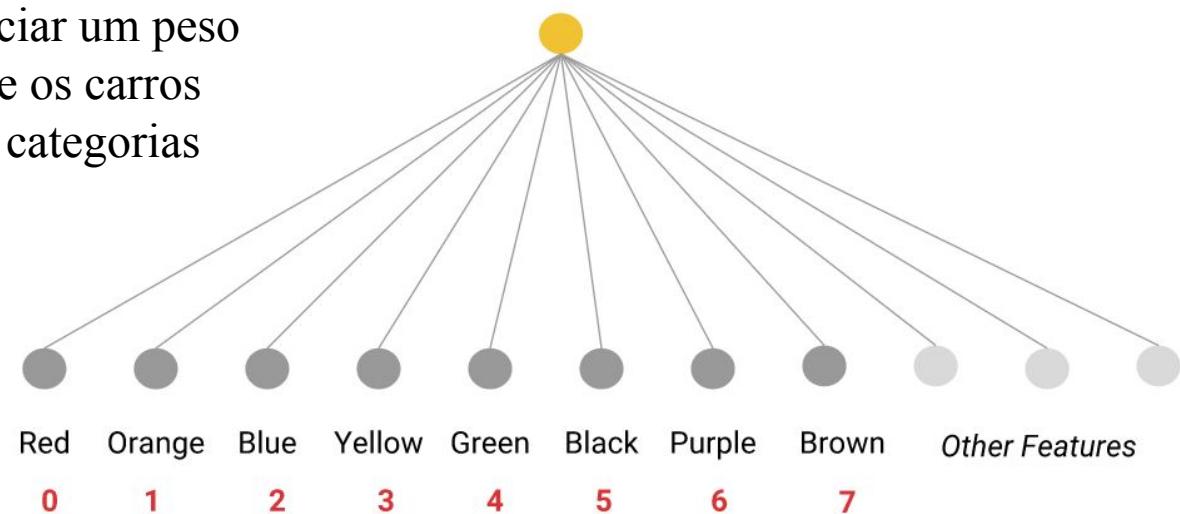
CATEGÓRICOS PARA BINÁRIO

- Pode-se representar valores categóricos como strings ou até mesmo números, mas não se pode comparar esses números ou subtraí-los uns dos outros.
- Se o número de categorias for pequeno, como o dia da semana ou uma paleta de cores limitada, pode-se criar um atributo binário para cada categoria. Por exemplo:

CATEGÓRICOS PARA RÁDIOS



Um especialista ou modelo de aprendizado pode associar um peso para cada cor. Por exemplo esse peso pode indicar que os carros vermelhos são mais caros do que os verdes. Assim as categorias são indexadas por esses pesos.



Fonte das figuras:

<https://developers.google.com/machine-learning/data-prep/transform/transform-categorical>

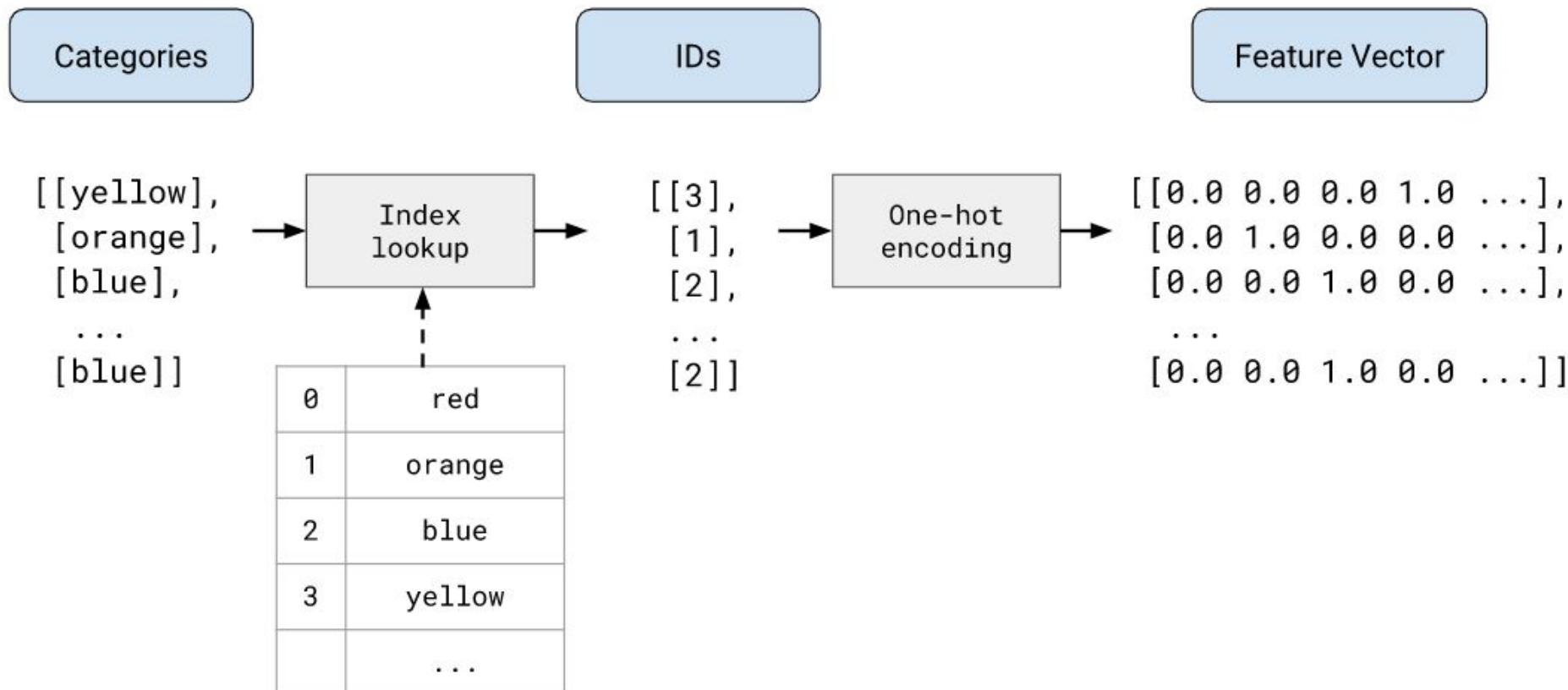
ONE-HOT-ENCODING

- one-hot-encoding: transforma **valores de atributos categóricos em atributos binários.**

	Cat	Dog	Zebra
	1	0	0
	0	1	0
	0	0	1

Fonte da figura: <http://www.stodolkiewicz.com/2019/12/16/one-hot-encoding/>

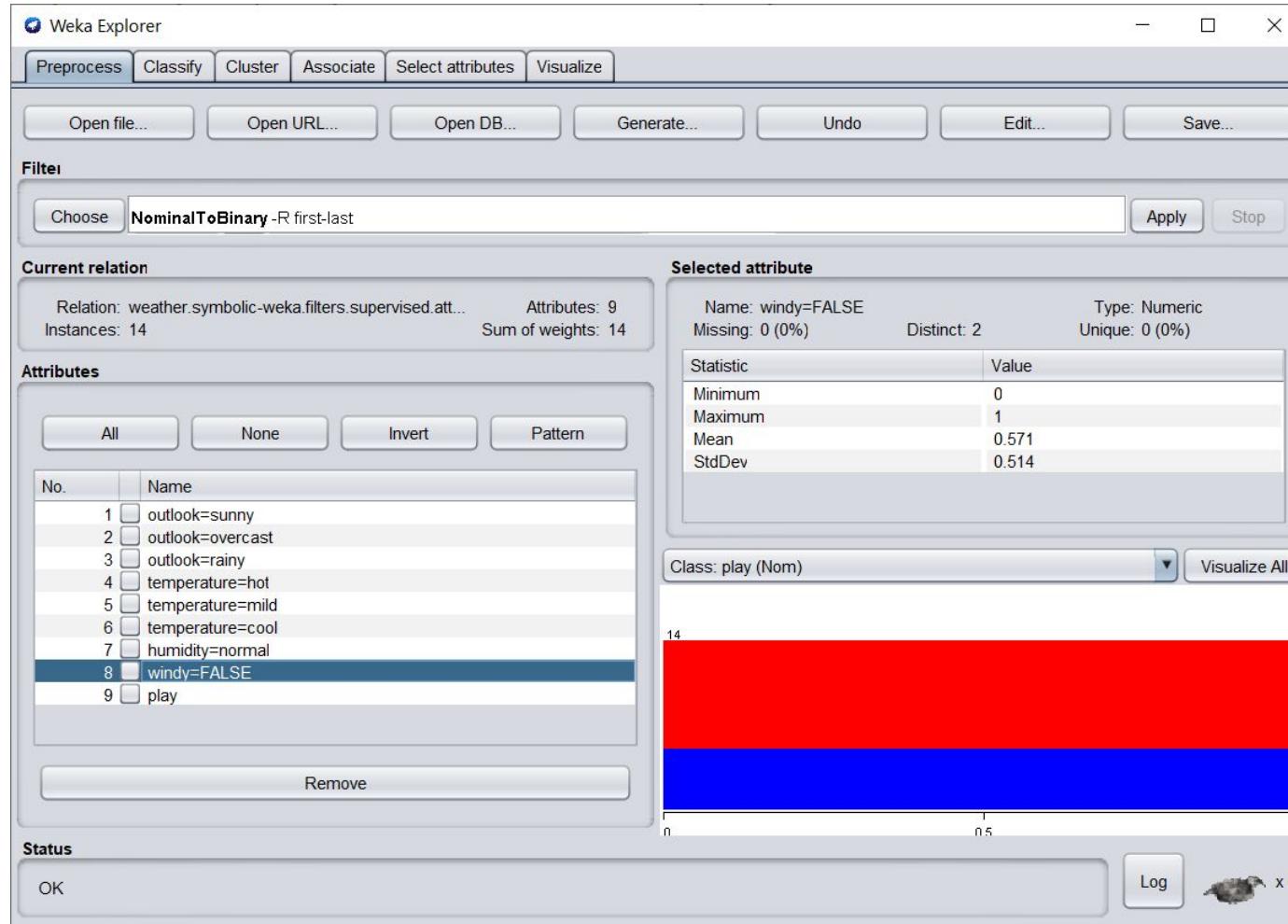
CATEGÓRICOS PARA BINÁRIOS



Fonte da figura:

<https://developers.google.com/machine-learning/data-prep/transform/transform-categorical>

EXEMPLO



ONE-HOT-ENCODING CÓDIGO EM PYTHON

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder

target_variable = np.array(['cat', 'dog', 'zebra', 'zebra'])

# One Hot Encode With scikit-learn -----
encoder = OneHotEncoder()
sk_one_hot_encoded = encoder.fit_transform(target_variable.reshape(-1, 1)).toarray()
# [[1. 0. 0.]
# [0. 1. 0.]
# [0. 0. 1.]
# [0. 0. 1.]]
print(sk_one_hot_encoded)
```

CATEGÓRICO PARA REAL

- Target Encoder
 - Suponha existir duas variáveis: uma categórica (x) e uma numérica (y).
 - Transforma-se x em uma variável numérica transportando valores de y
 - Uma maneira de fazer isso é calcular a média de y para cada nível de x .

$$enc_i = \text{mean}(y|x = i)$$

- problema: alguns grupos podem ser muito pequenos ou muito variáveis para serem confiáveis.

CATEGÓRICO PARA REAL

- Uma opção para resolver esse problema é definir um meio-termo entre a média do grupo e a média global de y :

$$enc_i = w_i \times \text{mean}(y|x = i) + (1 - w_i) \times \text{mean}(y)$$

- onde w_i está entre 0 e 1, dependendo de quão “representante” é a média do grupo.
- TargetEncoder, MEstimateEncoder e JamesSteinEncoder diferem com base em como eles definem w_i .
- No TargetEncoder, o peso depende do número de elementos do grupo e do parâmetro denominado “suavização”.
- Quando a suavização é 0, conta-se apenas com as médias do grupo. Então, à medida que a suavização aumenta, a média global torna-se cada vez mais pesada, levando a uma regularização mais forte.

CATEGÓRICOS

I Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Choose MergeInfrequentNominalValues -N 100 -R 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17

Apply Stop

Current relation:

Relation: german_credit-weka.filters.unsupervised.attribute... Attributes: 21 Instances: 1000 Sum of weights: 1000

Selected attribute

Name: purpose_merged_infrequent_values Type: Nominal Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	domestic appliance or...	202	202.0
2	new car	234	234.0
3	used car	103	103.0
4	furniture/equipment	181	181.0
5	radio/tv	280	280.0

Attributes

All None Invert Pattern

No. Name

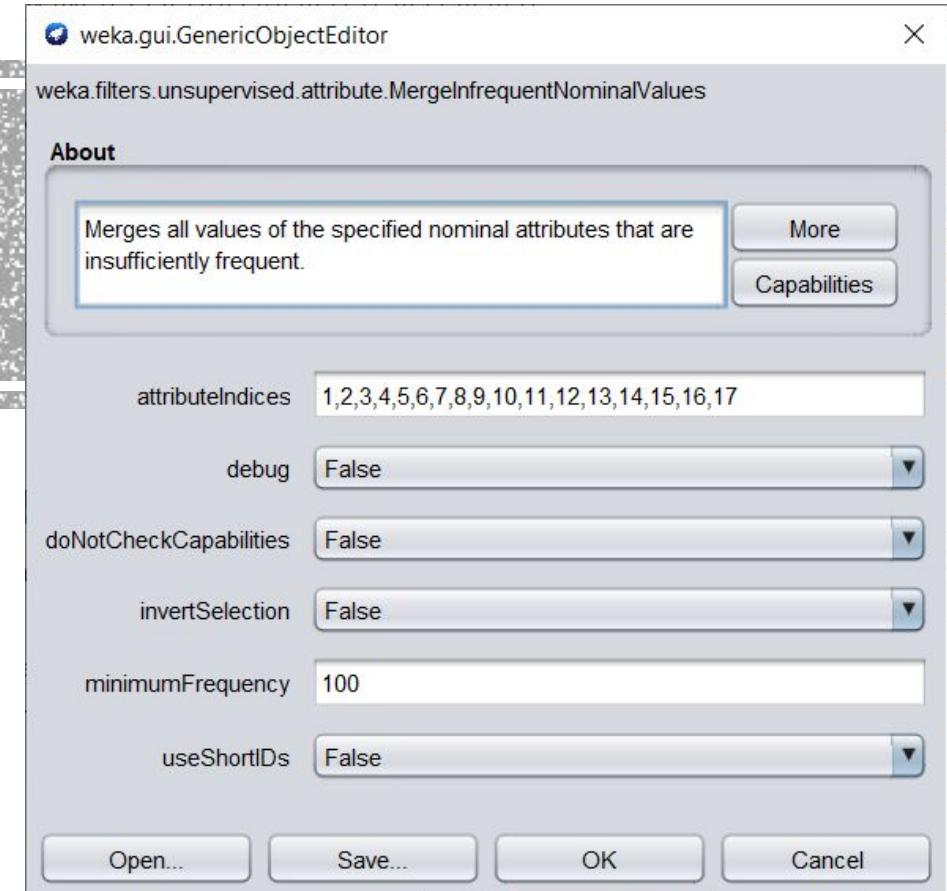
- checking_status
- duration
- credit_history_merged_infrequent_values
- purpose_merged_infrequent_values
- credit_amount
- savings_status_merged_infrequent_values
- employment
- installment_commitment
- personal_status_merged_infrequent_values
- other_parties_merged_infrequent_values
- residence_since
- property_magnitude
- age
- other_payment_plans
- housing
- existing_credits
- job
- number_of dependents

Remove

Status

OK

Class: class (Nom) Visualize All



- Quando existem categorias que são infreqüentes, sugere-se criar uma nova categoria que englobe as infreqüentes na análise
- Base credit-g.arrf
- MergeInfrequentNominalValues

NUMÉRICOS PARA

HIERARQUIA DE CONCEITOS

- Hierarquia de conceitos
 - Reduz os dados trocando-os por conceitos de maior nível semântico, ex. trocar o valor da idade por conceitos como jovem, adulto ou idoso.
 - Diferente da Discretização que reduz o número de valores de um atributo contínuo dividindo faixas de valores em intervalos ou valores discretos.



DISCRETIZAÇÃO

- Divide faixas de atributos contínuos em intervalos;
- Para que fazer discretização?
 - Porque alguns algoritmos de mineração só aceitam ou atributos categóricos ou atributos discretos
- Algumas técnicas:
 - Métodos de particionamento – equal-width (tamanho igual), equal-frequency (frequência igual)
 - Métodos baseados em entropia. Entropia: medida da quantidade de informação.



DISCRETIZAÇÃO POR INTERVALOS (BINNING)

- Valores de atributo (por exemplo, idade): 0, 4, 12, 16, 16, 18, 24, 26, 28

- Equi-width binning – para o bin de tamanho 10:

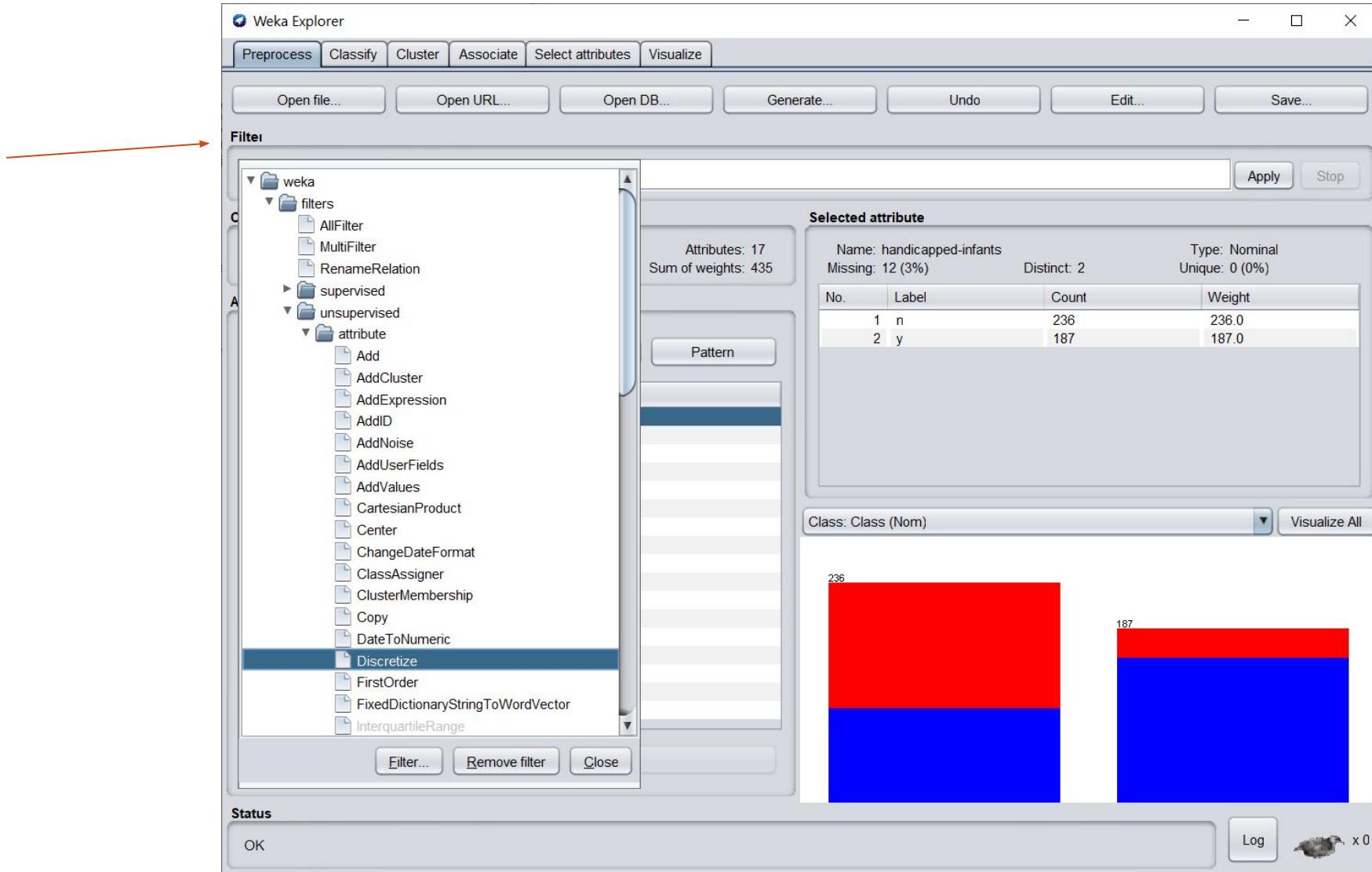
- Bin 1: 0, 4 [-,10) bin
 - Bin 2: 12, 16, 16, 18 [10,20) bin
 - Bin 3: 24, 26, 28 [20,+) bin
 - – infinito negativo, + infinito positivo

- Equi-frequency binning – para o bin com densidade de 3:

- Bin 1: 0, 4, 12 [-, 14) bin
 - Bin 2: 16, 16, 18 [14, 21) bin
 - Bin 3: 24, 26, 28 [21,+] bin



EXEMPLO



EXEMPLO

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose Discretize -B 10 -M 1.0 -R first-last -precision 6

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Discretize-B1... Attributes: 5
Instances: 150 Sum of weights: 150

Attributes

All None Invert Pattern

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Remove

Selected attribute

Name: sepallength Missing: 0 (0%) Distinct: 10 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-4.66]	9	9.0
2	'(4.66-5.02]	23	23.0
3	'(5.02-5.38]	14	14.0
4	'(5.38-5.74]	27	27.0
5	'(5.74-6.1]	22	22.0
6	'(6.1-6.46]	20	20.0
7	'(6.46-6.82]	18	18.0
8	'(6.82-7.18]	6	6.0
9	'(7.18-7.54]	5	5.0
10	'(7.54,inf]	6	6.0

Class: class (Nom) Visualize All

9 23 14 27 22 20 18 6 5 6

Status

OK

weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More Capabilities

attributIndices first-last

binRangePrecision 6

bins 10 **Equi-width binning**

debug False

endWeightOfInstancesPerInterval -1.0

doNotCheckCapabilities False

findNumBins False

ignoreClass False

invertSelection False

makeBinary False

spreadAttributeWeight False

useBinNumbers False

useEqualFrequency False

Open... Save... OK Cancel

EXEMPLO

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose Discretize -Y -F -B 5 -M -1.0 -R first-last -precision 6 Apply Stop

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Discretize-Y-F.. Attributes: 5 Instances: 150 Sum of weights: 150

Attributes

All None Invert Pattern

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Remove

Status OK

Selected attribute

Name: sepallength Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	'B1of5'	32	32.0
2	'B2of5'	27	27.0
3	'B3of5'	30	30.0
4	'B4of5'	31	31.0
5	'B5of5'	30	30.0

Class: class (Nom) Visualize All

Log x 0

weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

More Capabilities

attributeIndices first-last

binRangePrecision 6

bins 5

debug False

desiredWeightOfInstancesPerInterval -1.0

doNotCheckCapabilities False

findNumBins False

ignoreClass False

invertSelection False

makeBinary False

spreadAttributeWeight False

useBinNumbers True

useEqualFrequency True

Equi-frequency binning

Open... Save... OK Cancel

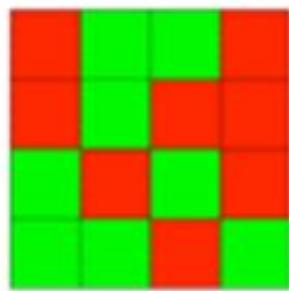
DISCRETIZAÇÃO USANDO ENTROPIA

- Ordene os valores dos atributos e agregue a ele a sua classe:
 - (0,P), (4,P), (12,P), (16,N), (16,N), (18,P), (24,N), (26,N), (28,N)
- Criação de intervalos com base na entropia:
 - Intuitivo: encontre a melhor divisão que os bins são os mais puros o possível;
 - Formalmente caracterizado pelo maior ganho de informação.
 - Seja S os 9 pares acima, $p=4/9$ é a fração de pares P , e $n=5/9$ é a fração de Pares N.
 - Entropia(S) = $- p \log_2 p - n \log_2 n$.
 - Entropia pequena – o conjunto é relativamente puro; valor próximo de 0.
 - Entropia grande – o conjunto é heterogêneo; valor próximo de 1.

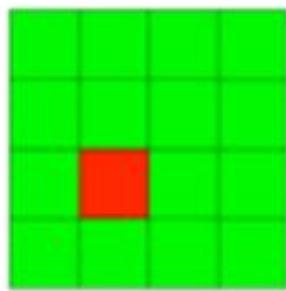


ENTROPIA

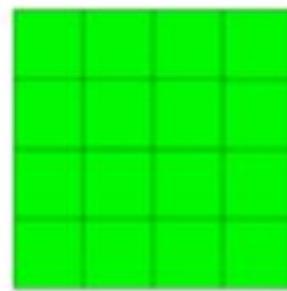
é um jeito de medir a uniformidade da distribuição



$$H = 1.0 \text{ bits}$$



$$H = 0.3 \text{ bits}$$



$$H = 0.0 \text{ bits}$$



DISCRETIZAÇÃO USANDO ENTROPIA

- Seja v um possível ponto de corte do intervalo. Então S é dividido em dois conjuntos:
 - S_1 : com valores $\leq v$ e S_2 : com valores $> v$
- Informação da divisão:
 - $I(S_1, S_2) = (|S_1|/|S|) \text{ Entropia}(S_1) + (|S_2|/|S|) \text{ Entropia}(S_2)$
- Ganho de Informação na divisão:
 - $\text{Ganho}(v, S) = \text{Entropia}(S) - I(S_1, S_2)$
- Objetivo: dividir atingindo o máximo de ganho de informação.
 - Possível ponto de corte: $v = 14$.



DISCRETIZAÇÃO BASEADA NA ENTROPIA

- Para v=14,
 - $I(S1, S2) = 0 + 6/9 * \text{Entropia}(S2) = 6/9 * 0.65 = 0.433$
 - $\text{Ganho}(14, S) = \text{Entropia}(S) - 0.433 = (- (4/9) (\log (4/9)) - (5/9) \log (5/9)) - 0.433 = 0.99 - 0.43 = 0,56$
- Máximo ganho Gain significa mínimo I.
- O melhor critério de divisão é encontrado após examinar todas as possibilidades.

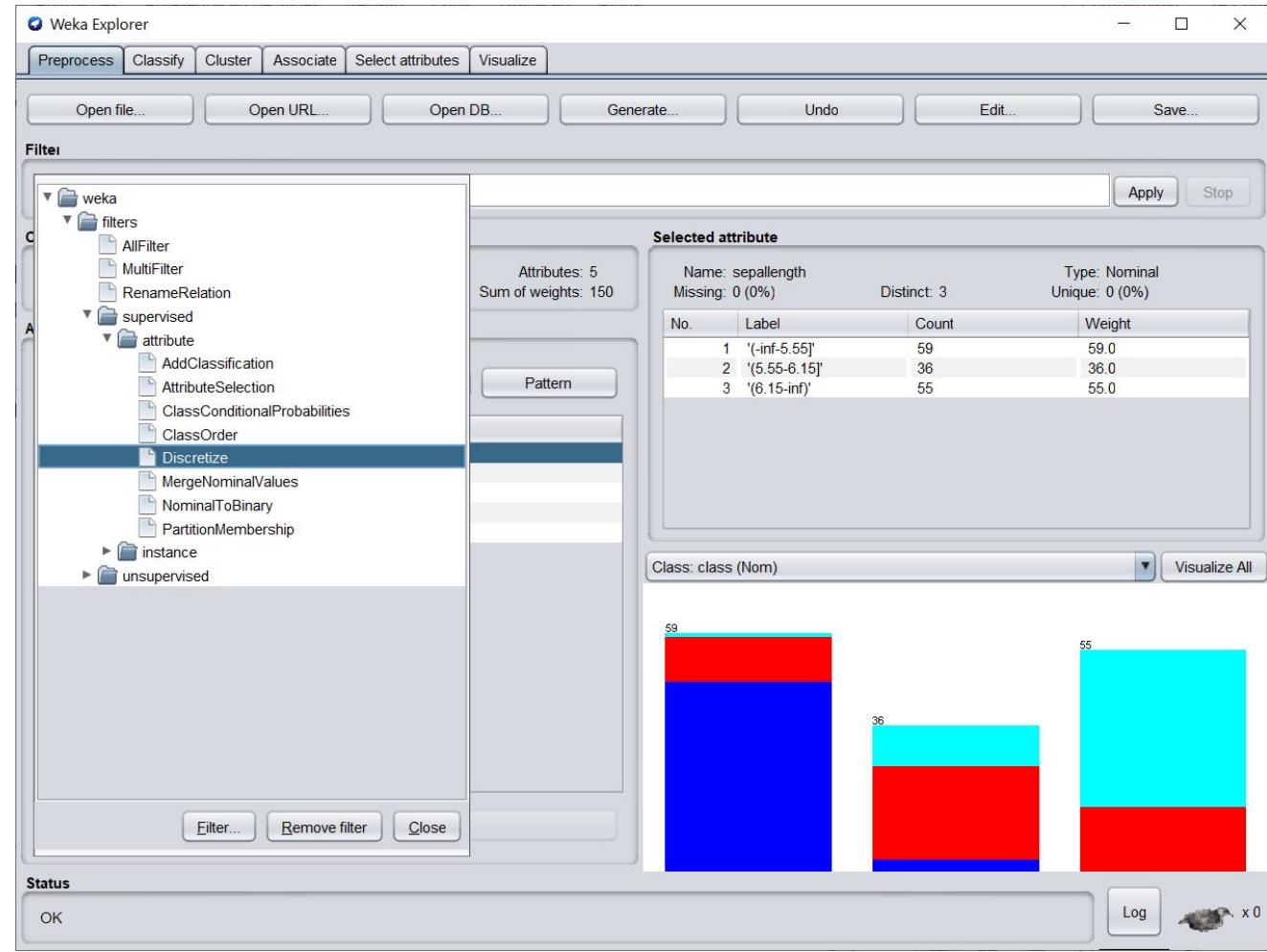


DISCRETIZADORES DESENVOLVIDOS

- Omega – corta dado em intervalos de acordo com mudança de labels e faz junção de intervalos de granularidade fina de acordo com critério de “erro”;
- UseMiner – faz discretização usando padronização, por faixas de desvio padrão;

EXEMPLO

Discretização baseada na entropia

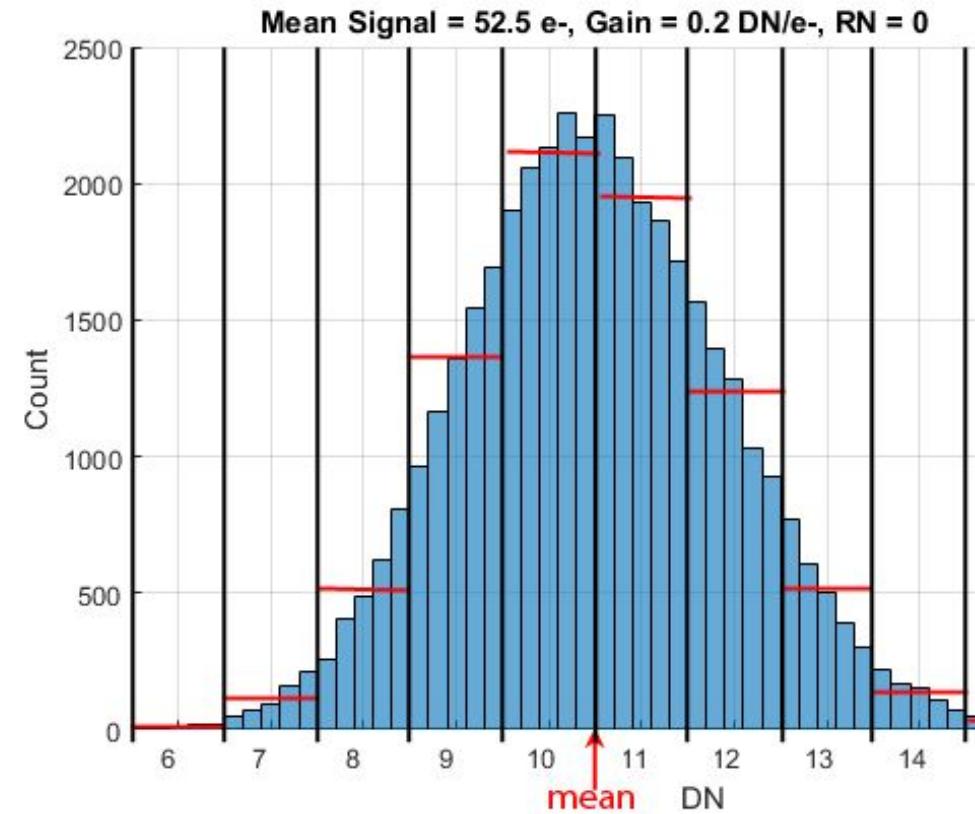


REDUZINDO O CONJUNTO DE VALORES

quantização

QUANTIZAÇÃO O QUE É?

- Quantização é um termo abrangente que cobre muitas técnicas diferentes para converter um grande conjunto de valores de entrada para um conjunto menor de saída.
- Diminui-se o domínio do atributo, otimizando o processo de mineração que passa a trabalhar com um domínio menor de dados por atributo.
- Analogia. São 11:30, mas não estou sendo 100% precisa, pois não informei os segundos e os milésimos de segundo.



Fonte da Figura: <https://www.strollswithmydog.com/sub-lsb-quantization/>

QUANTIZAÇÃO DE IMAGENS

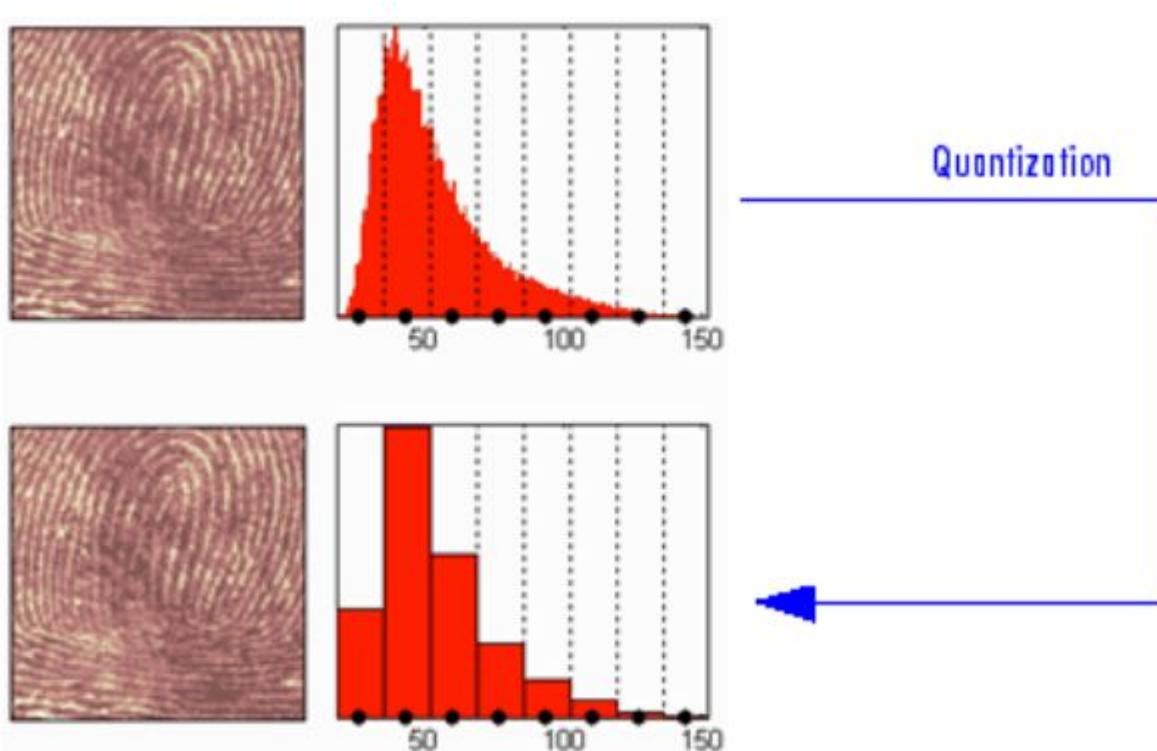
- Uma cor em uma imagem digital pode ser representada por 24 bits (true color). Mas esse valor pode ser quantizado. A quantização, em essência, diminui o número de bits necessários para representar a informação.



24 bits per pixel

Fonte da Figura: <https://www.qualcomm.com/news/onq/2019/03/12/heres-why-quantization-matters-ai>

QUANTIZAÇÃO



Determine a largura do bin:

- associe a cada bin a média (ou o máximo ou a mediana) dos valores originais

Fonte da Figura: <https://jp.mathworks.com/>

DISCUSSÃO

- Para que?
 - mapear dados de categóricos para inteiros
 - discretizar
 - quantizar
- Você já fez isso? Em qual situação e por que?