# AceleraDev Data Science

Engenharia de Features

# Variáveis categóricas

/

- Sem aplicações matemáticas;
- Agrupadores, classificadores;
- Possuem um limite de valores;
- "Encode"
  - Label Encode
  - One Hot Encode

## Label Encoding

| Food Name | Categorical # | Calories |
|---|---|---|
| Apple | 1 | 95 |
| Chicken | 2 | 231 |
| Broccoli | 3 | 50 |

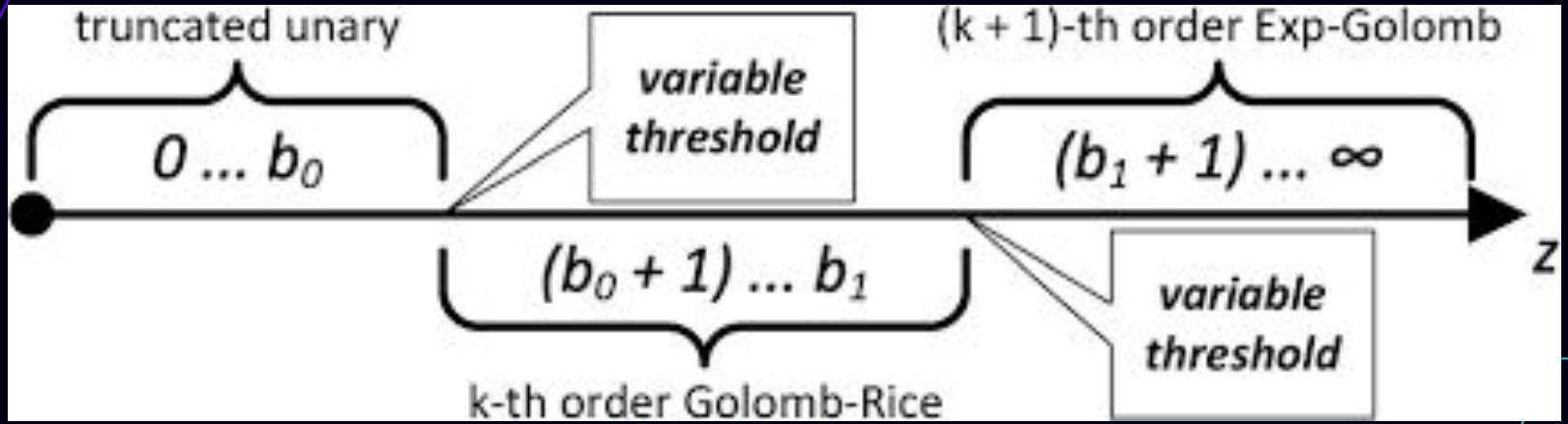
## One Hot Encoding

| Apple | Chicken | Broccoli | Calories |
|---|---|---|---|
| 1 | 0 | 0 | 95 |
| 0 | 1 | 0 | 231 |
| 0 | 0 | 1 | 50 |

# Binarização

/

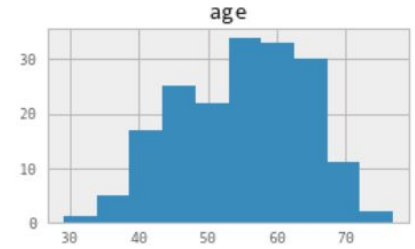
- Transforma valores escalares em **binários.**
- Por padrão tudo que é positivo recebe valor **1.**
- Boa prática **normalizar/padronizar os valores.**

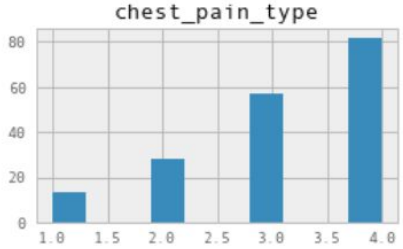

truncated unary
0 ... $b_0$
variable threshold
(k + 1)-th order Exp-Golomb
$(b_1 + 1)$ ... ∞
z
$(b_0 + 1)$ ... $b_1$
variable threshold
k-th order Golomb-Rice

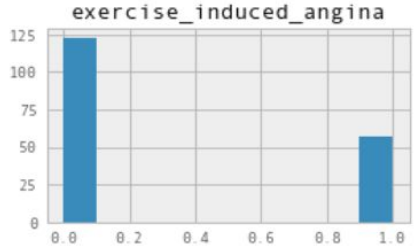
# Quantização(Binning)

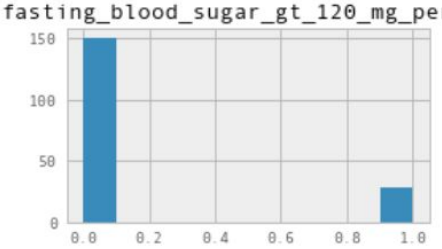
/

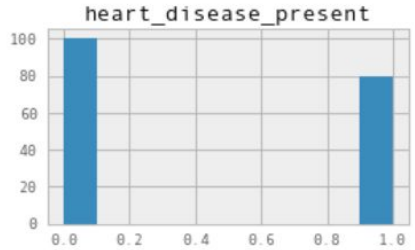
- Separa as amostras em quartis de quantidade iguais.
- *bins:* Quantidade de  "separações".
- Permite o agrupamento e criação de *ranges*

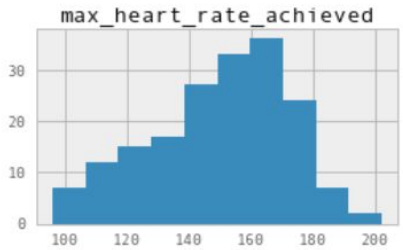

age
30
20
10
0
30
40
50
60
70

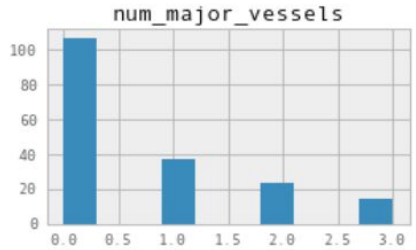

chest_pain_type
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

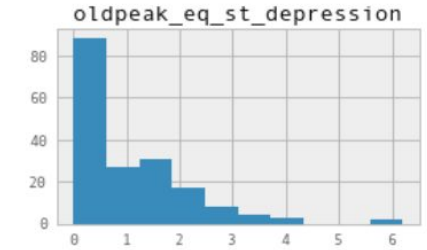

exercise_induced_angina
125
100
75
50
25
0
0.0
0.2
0.4
0.6
0.8
1.0

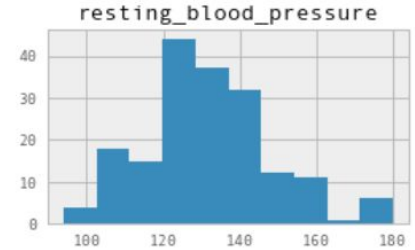

fasting_blood_sugar_gt_120_mg_per_dl
150
100
50
0
0.0
0.2
0.4
0.6
0.8
1.0

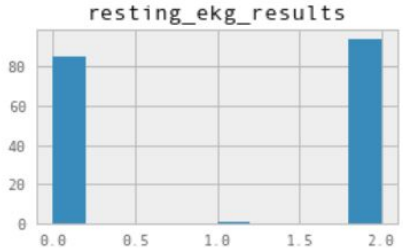

heart_disease_present
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0

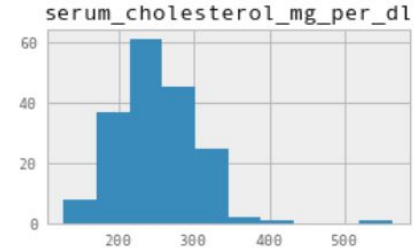

max_heart_rate_achieved
30
20
10
0
100
120
140
160
180
200

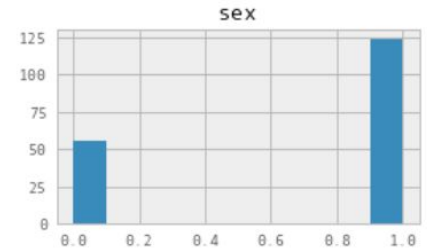

num_major_vessels
100
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

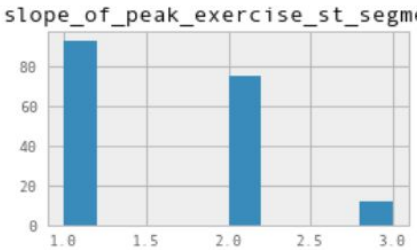

oldpeak_eq_st_depression
80
60
40
20
0
0
1
2
3
4
5
6


resting_blood_pressure
40
30
20
10
0
100
120
140
160
180


resting_ekg_results
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0


serum_cholesterol_mg_per_dl
60
40
20
0
200
300
400
500


sex
125
100
75
50
25
0
0.0
0.2
0.4
0.6
0.8
1.0


slope_of_peak_exercise_st_segment
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0

# AceleraDev Data Science

Engenharia de Features

# Variáveis Números

/

- StandardScaler
  - Z-score
  - (X - Média) / (Desvio Padrão)

# Variáveis Númericas

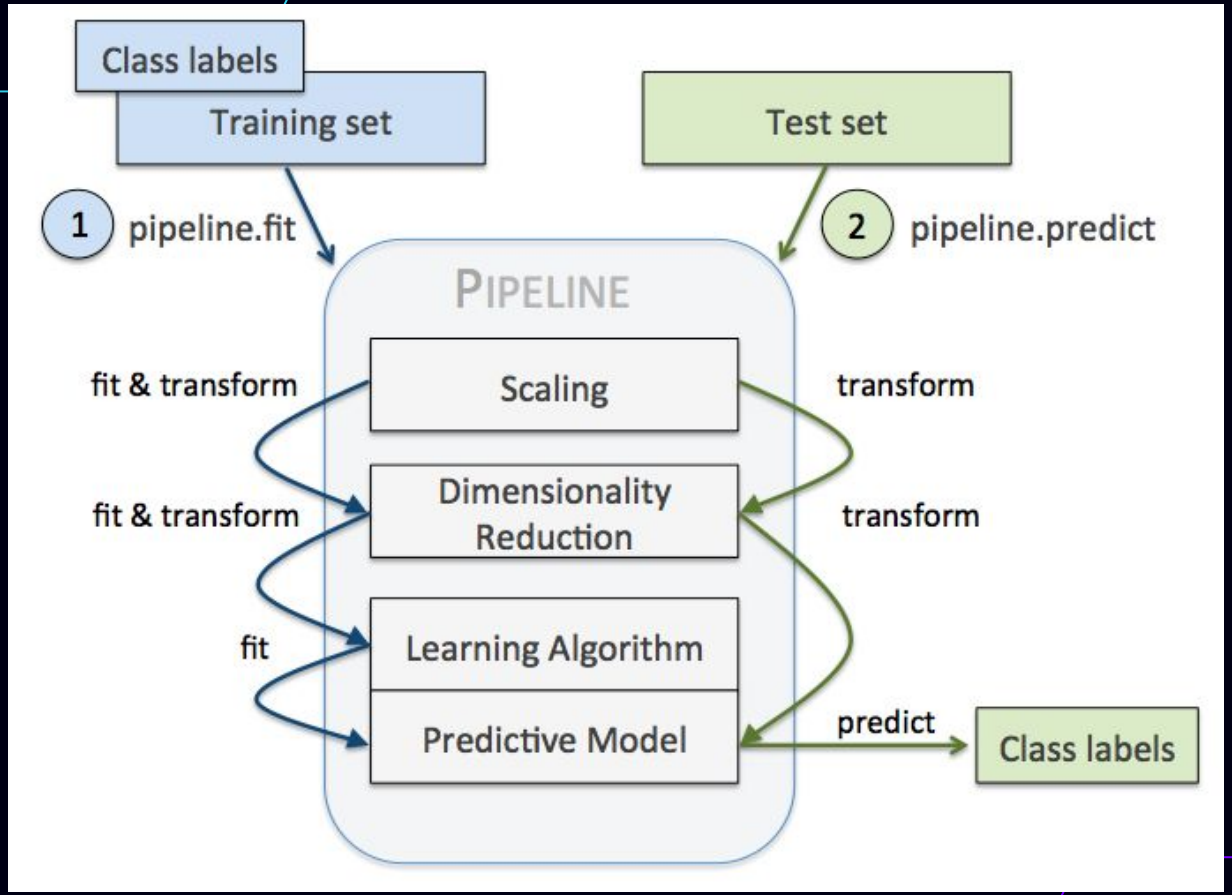
/

- `MinMaxScaler`
  - X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
  - X_scaled = X_std * (max - min) + min

# Normalização, Escala Transformação

/

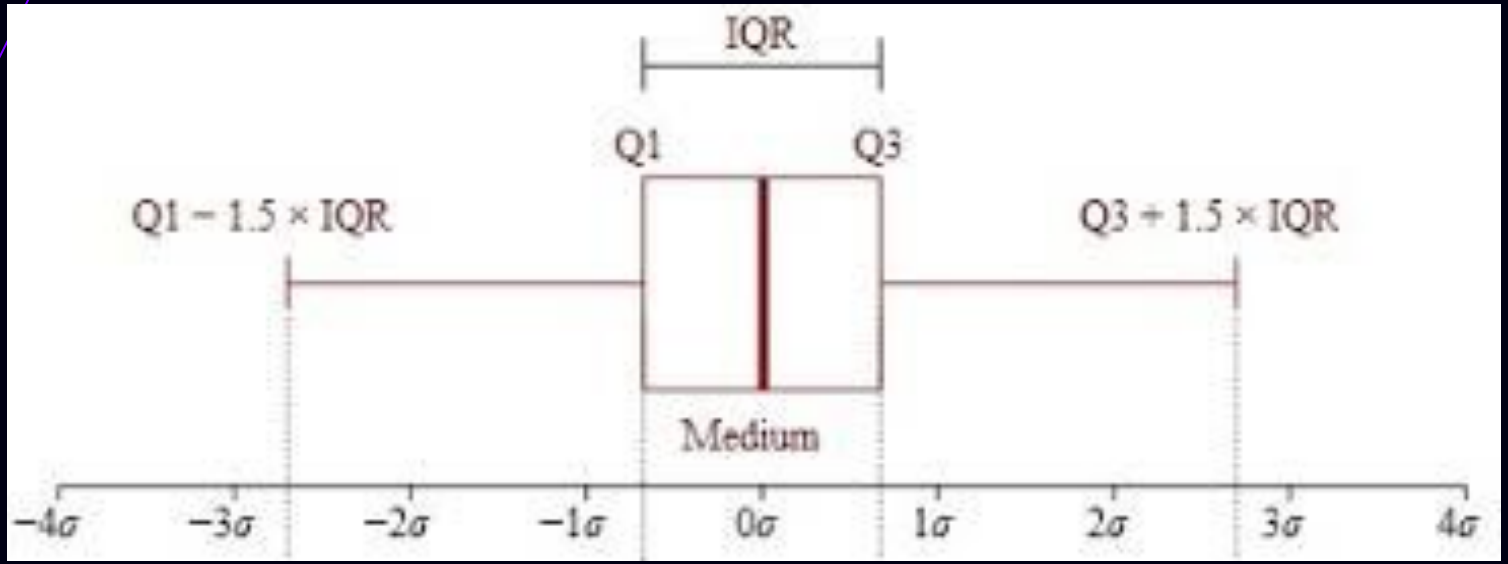
- Aplicação de técnicas de normalização dos dados, de forma automática.
  - Preenchimento de valores faltantes;
  - Aplicação de Padronização e Normalização;
  - Transformação de colunas, por valores ou encode.
- O agrupamento permite a criação de *pipelines*
  - Reaproveitamento dos cálculos;
  - Aplicação simultânea


Class labels
Training set
Test set
1
pipeline.fit
2
pipeline.predict
PIPELINE
fit & transform
Scaling
transform
fit & transform
Dimensionality Reduction
transform
fit
Learning Algorithm
Predictive Model
predict
Class labels

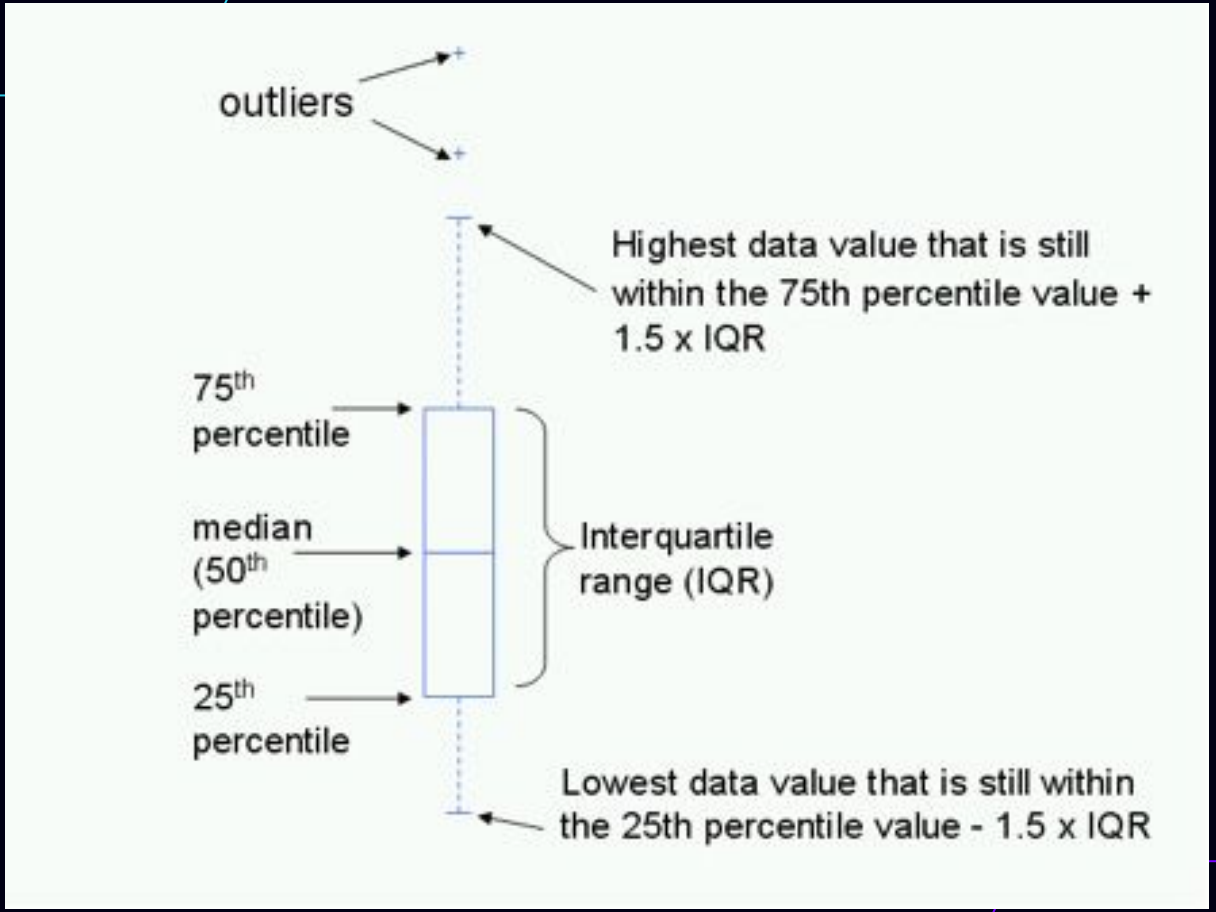
# Remoção de outliers

/

- Procurando por outliers
- IQR
  - Menores que o Q1;
  - Maiores que o Q3;
  - Fórmula
    - IQR: Q3-Q1;
    - Abaixo: Q1 - **x***IQR;
    - Acima: Q3 + **x***IQR;
    - **x** padrão 1,5




$$\mathrm{IQR} = Q_3 - Q_1$$


outliers
Highest data value that is still within the 75th percentile value + 1.5 x IQR
75th percentile
median (50th percentile)
25th percentile
Interquartile range (IQR)
Lowest data value that is still within the 25th percentile value - 1.5 x IQR

# Outliers Formula

**Lower Outlier = Q1 – (1.5 x IQR)**

**Higher Outlier = Q3 + (1.5 x IQR)**

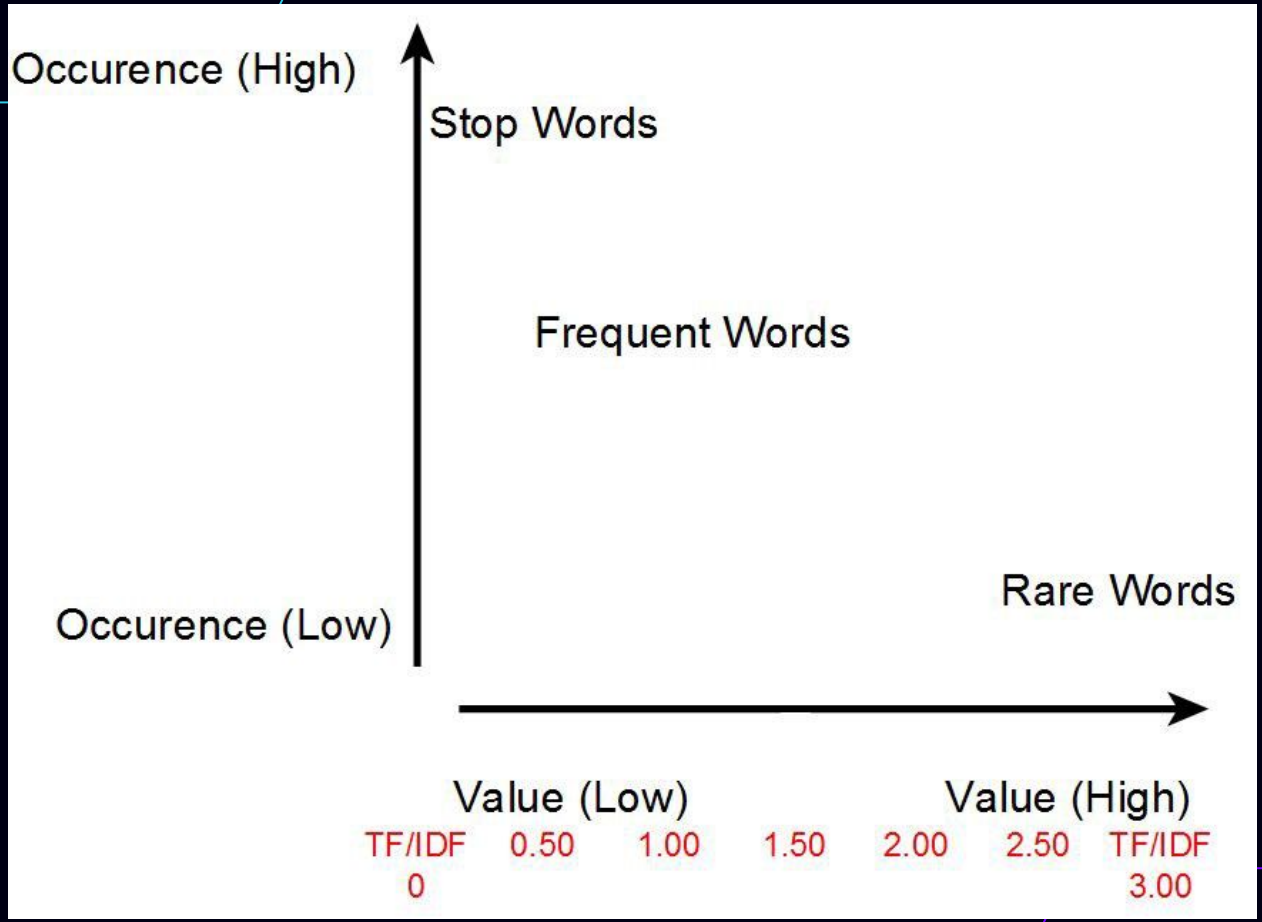
# AceleraDev Data Science

Engenharia de Features

# Features de Texto

/

- Categóricas por padrão
- Contagem
  - Palavras, expressões,
- TF-IDF
  - Frequência do Termo
    - O peso de um termo que ocorre em um documento é diretamente proporcional à sua frequência
  - Inverso da Frequência no documento
    - A especificidade de um termo pode ser quantificada por uma função inversa do número de documentos em que ele ocorre


Occurence (High)
Stop Words
Frequent Words
Rare Words
Occurence (Low)
Value (Low)
Value (High)
TF/IDF
0
0.50
1.00
1.50
2.00
2.50
TF/IDF
3.00

# TFIDF

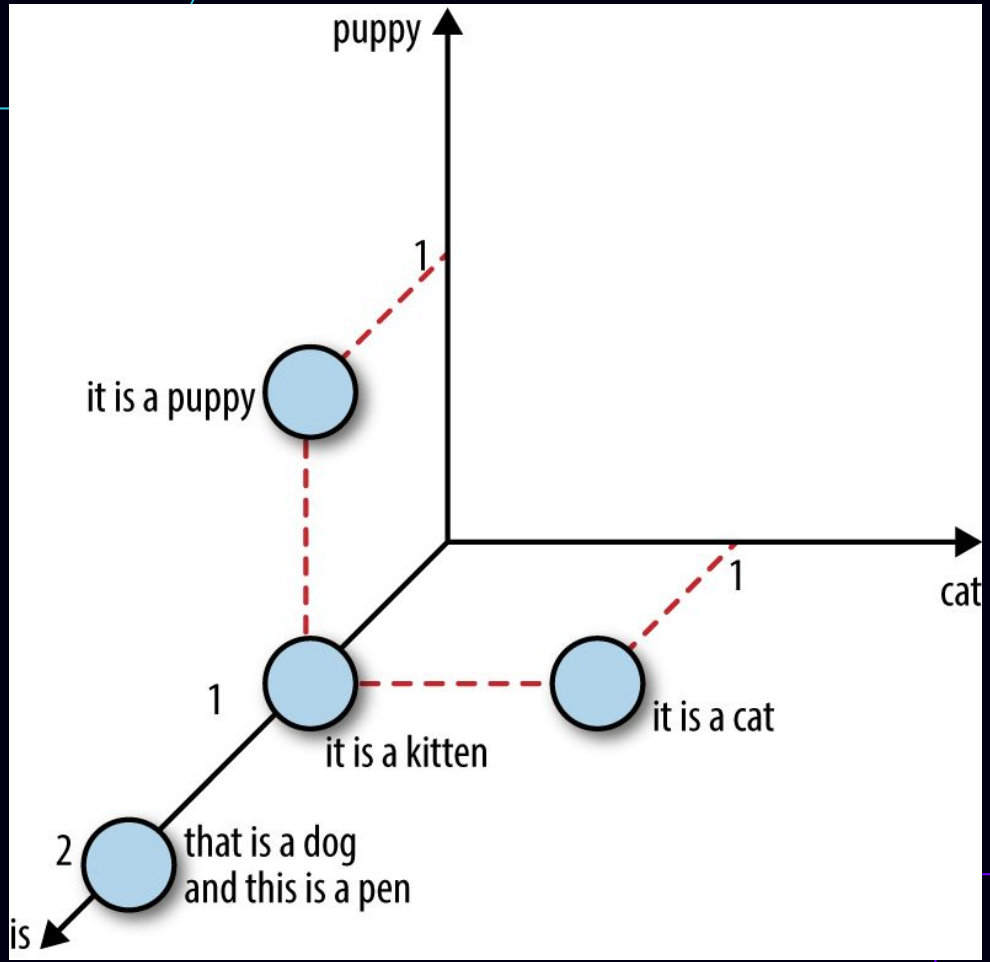
For a term $i$ in document $j$:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{ij}$ = number of occurrences of $i$ in $j$

$df_i$ = number of documents containing $i$

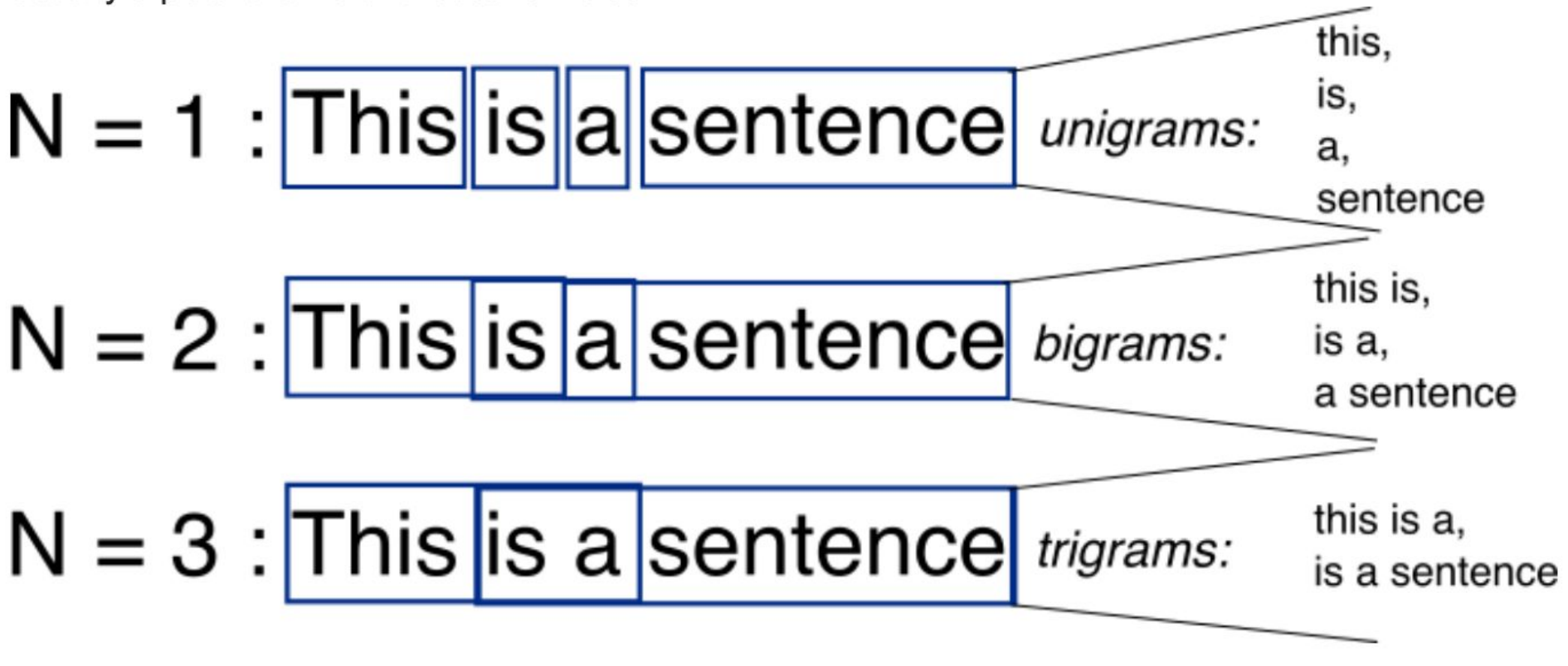
$N$ = total number of documents


puppy
1
it is a puppy
1
cat
1
it is a cat
it is a kitten
2
that is a dog
and this is a pen
is

# Features de Texto

/

- Agrupamento de termos;\
- Ordenação
- N-gram:
  - Agrupamento de 2 ou mais palavras, onde o a partir deste agrupamento passam a ser tratadas com 1 elemento único
- Stop-words:
  - palavras ou composições que não impactam em valor e entendimento.Normalmente são palavras de ligação como artigos, advérbios, sufixos, prefixos e radicais


N = 1 : This is a sentence
unigrams:
this,
is,
a,
sentence
N = 2 : This is a sentence
bigrams:
this is,
is a,
a sentence
N = 3 : This is a sentence
trigrams:
this is a,
is a sentence

```
> stopwords("english")
  [1] "i"          "me"         "my"         "myself"     "we"
  [6] "our"        "ours"       "ourselves"  "you"        "your"
 [11] "yours"      "yourself"   "yourselves" "he"         "him"
 [16] "his"        "himself"    "she"        "her"        "hers"
 [21] "herself"    "it"         "its"        "itself"     "they"
 [26] "them"       "their"      "theirs"     "themselves" "what"
 [31] "which"      "who"        "whom"       "this"       "that"
 [36] "these"      "those"      "am"         "is"         "are"
 [41] "was"        "were"       "be"         "been"       "being"
 [46] "have"       "has"        "had"        "having"     "do"
 [51] "does"       "did"        "doing"      "would"      "should"
 [56] "could"      "ought"      "i'm"        "you're"     "he's"
 [61] "she's"      "it's"       "we're"      "they're"    "i've"
 [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
 [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"
 [76] "you'll"     "he'll"      "she'll"     "we'll"      "they'll"
 [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"
 [86] "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"
 [96] "cannot"     "couldn't"   "mustn't"    "let's"      "that's"
[101] "who's"      "what's"     "here's"     "there's"    "when's"
[106] "where's"    "why's"      "how's"      "a"          "an"
```

# Feature Engineering

/

- https://jorisvandenbossche.github.io/blog/2018/05/28/scikit-learn-columntransformer
- https://medium.com/vickdata/easier-machine-learning-with-the-new-column-transformer-from-scikit-learn-c2268ea9564c
- https://code-examples.net/pt/docs/scikit_learn/modules/impute
- https://medium.com/datadriveninvestor/finding-outliers-in-dataset-using-python-efc3fce6ce32
- https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
- https://adataanalyst.com/scikit-learn/countvectorizer-sklearn-example/
- https://towardsdatascience.com/hacking-scikit-learns-vectorizers-9ef26a7170af
  https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789808452/1/ch01lvl1sec17/binarization