

Pré-processamento dos dados

Amostragem de dados

Amostragem

- Amostra pequena tende a não representar bem o problema
- Amostra muito grande tem custo alto
- Ideal: amostra não grande com dados que refletem a distribuição estatística real
- Não há garantia, mas há abordagens que aumentam esta chance
 - Amostragem aleatória simples
 - Amostragem estratificada
 - Amostragem progressiva

Amostragem aleatória simples

- Sem reposição: cada instância pode ser selecionada apenas uma vez
- Com reposição: probabilidade de escolher qualquer instância se mantém constante

Amostragem estratificada

- Adequada para classes com propriedades diferentes
 - Ex: número de instância bem diferentes
- Manter o mesmo número de instância para cada classe; OU
- Respeitar a proporcionalidade original

Amostragem progressiva

- Começa com uma amostra pequena e aumenta progressivamente, enquanto a acurácia preditiva continua a melhorar
- Resultado: é possível definir a menor quantidade de dados necessária
- Geralmente fornece uma boa estimativa para o tamanho da amostra

Dados faltantes

Leitura de dataset

```
1 import pandas as pd
2 from io import StringIO
3
4 csv_data = '''A,B,C,D
5 1.0,2.0,3.0,4.0
6 5.0,6.0,,8.0
7 10.0,11.0,12.0,''''
8
9 # If you are using Python 2.7, you need
10 # to convert the string to unicode:
11 # csv_data = unicode(csv_data)
12
13 df = pd.read_csv(StringIO(csv_data))
14 df
```

	A	B	C	D
0	1	2	3	4
1	5	6	NaN	8
2	10	11	12	NaN

(I) Eliminação de exemplos

`df.dropna()`

	A	B	C	D
0	1	2	3	4
1	5	6	NaN	8
2	10	11	12	NaN

(II) Eliminação de características

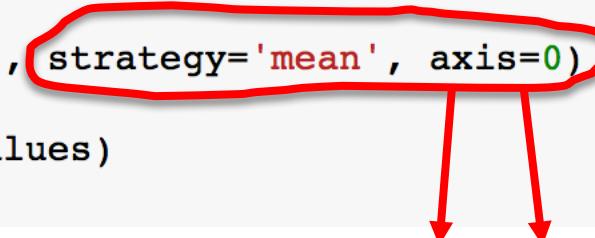
df.dropna(axis=1)

	A	B	C	D
0	1	2	3	4
1	5	6	NaN	8
2	10	11	12	NaN

(III) Uso de técnicas de interpolação

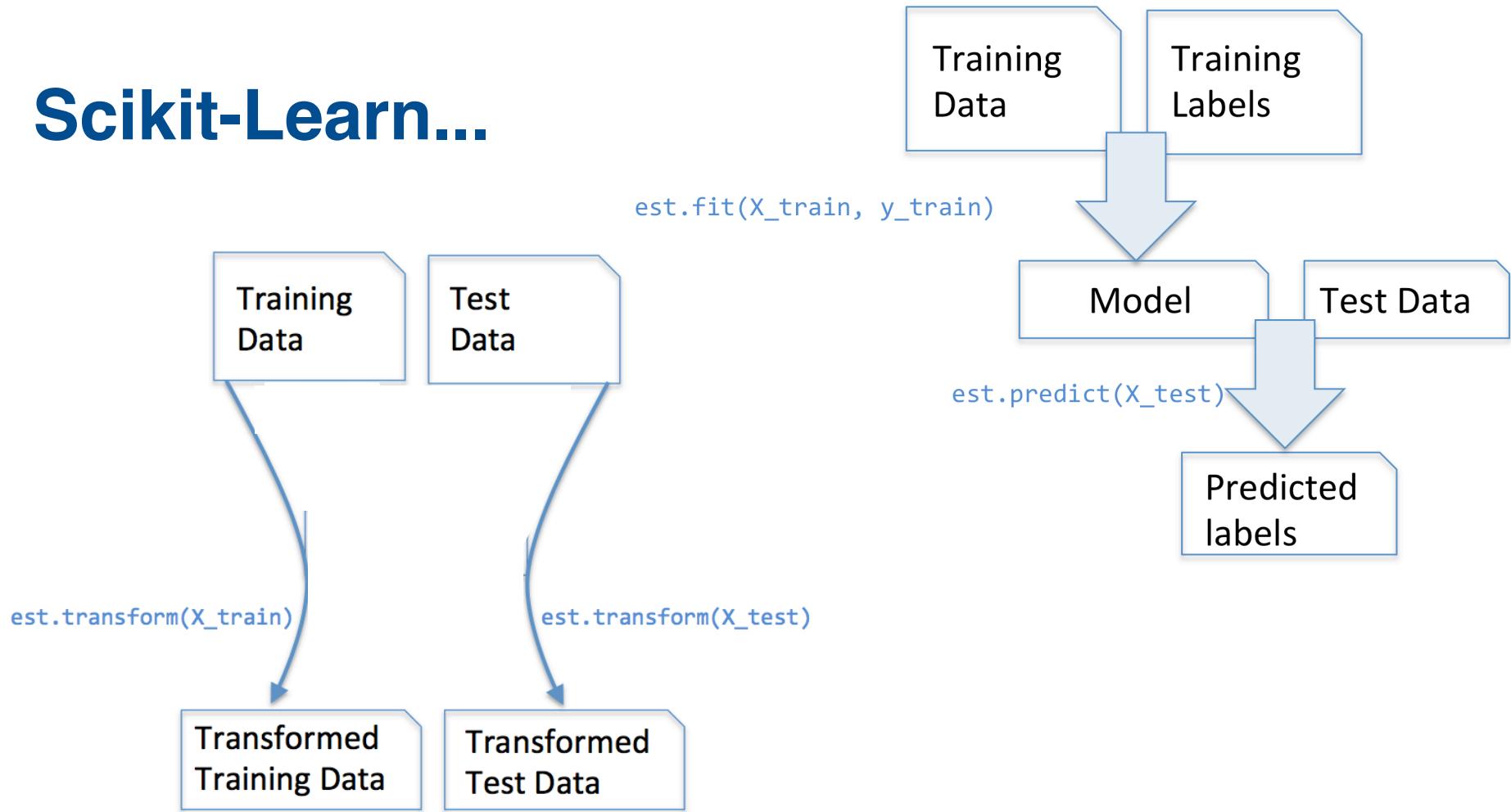
Estimar valores faltantes a partir dos outros valores presentes no dataset

```
1 from sklearn.preprocessing import Imputer  
2  
3 imr = Imputer(missing_values='NaN', strategy='mean', axis=0)  
4 imr = imr.fit(df)  
5 imputed_data = imr.transform(df.values)  
6 imputed_data
```



	A	B	C	D
0	1	2	3	4
1	5	6	7.5	8
2	10	11	12	6

Scikit-Learn...



Dados inconsistentes

(I) Conflito Features vs. Rótulos

Idade	Sexo	Peso	Manchas	Temp.	# Int.	Diagnóstico
28	M	79	Concentradas	38,0	2	Doente
18	F	67	Inexistentes	39,5	4	Doente
49	M	92	Espalhadas	38,0	2	Saudável
18	M	43	Inexistentes	38,5	8	Doente
21	F	52	Uniformes	37,6	1	Saudável
22	F	72	Inexistentes	38,0	3	Doente
19	F	87	Espalhadas	39,0	6	Doente
22	F	72	Inexistentes	38,0	3	Saudável



Entre todos os atributos de entrada e o atributo de saída

- Solução: algoritmos de verificação automática para eliminação das entradas inconsistentes

(II) Conflito entre features

Idade	Sexo	Peso	Manchas	Temp.	# Int.	Diagnóstico
28	M	79	Concentradas	38,0	2	Doente
18	F	67	Inexistentes	39,5	4	Doente
49	M	92	Espalhadas	38,0	2	Saudável
18	M	43	Inexistentes	38,5	8	Doente
21	F	52	Uniformes	37,6	1	Saudável
22	F	72	Inexistentes	38,0	3	Doente
19	F	87	Espalhadas	39,0	6	Doente
22	F	72	Inexistentes	38,0	3	Saudável
2	F	87	Espalhadas	39,0	6	Doente

Entre atributos de entrada

- Solução: algoritmos de verificação automática com conhecimento especialista
 - Ex: Se Peso > 20 * Idade...

Dados redundantes

(I) Instâncias redundantes

Idade	Sexo	Peso	Manchas	Temp.	# Int.	Diagnóstico
28	M	79	Concentradas	38,0	2	Doente
18	F	67	Inexistentes	39,5	4	Doente
49	M	92	Espalhadas	38,0	2	Saudável
18	F	67	Inexistentes	39,5	4	Doente
18	M	43	Inexistentes	38,5	8	Doente
21	F	52	Uniformes	37,6	1	Saudável
22	F	72	Inexistentes	38,0	3	Doente
19	F	87	Espalhadas	39,0	6	Doente
34	M	67	Uniformes	38,4	2	Saudável

Instâncias redundantes
participam mais de uma vez
do processo de ajuste de
parâmetros de um modelo,
contribuindo dessa forma, mais que
outras instâncias para a definição do
modelo final

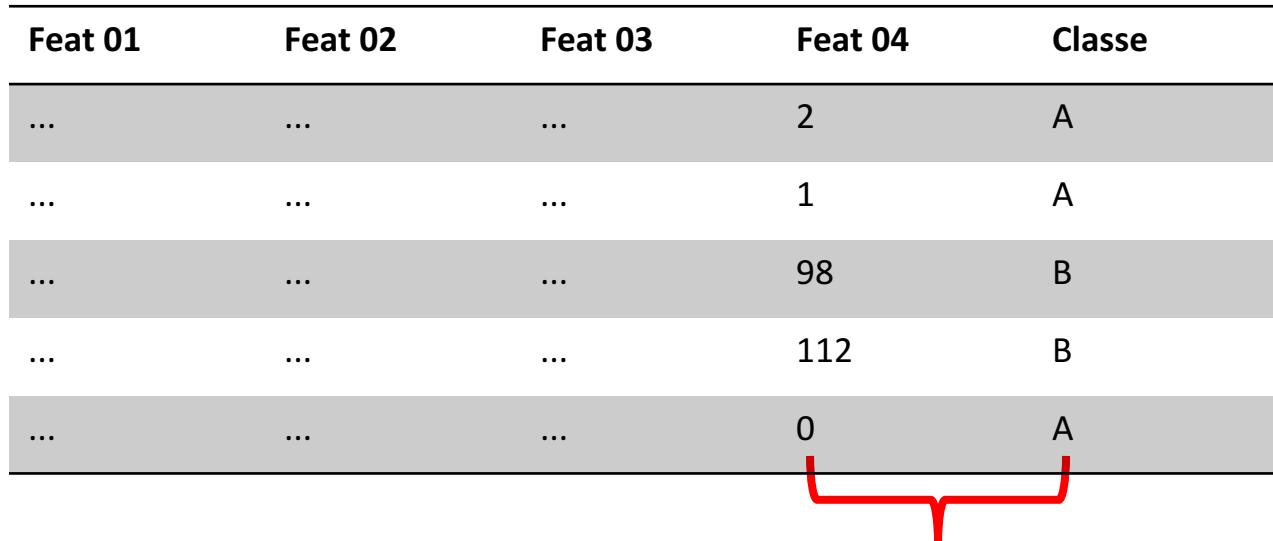
- Solução: eliminação de n-1 redundâncias

(II) Features redundantes

Idade	Sexo	Peso	Manchas	Temp.	# Int.	# Vis.	Diagnóstico
28	M	79	Concentradas	38,0	2	2	Doente
18	F	67	Inexistentes	39,5	4	4	Doente
49	M	92	Espalhadas	38,0	2	2	Saudável
18	M	43	Inexistentes	38,5	8	8	Doente
21	F	52	Uniformes	37,6	1	1	Saudável
22	F	72	Inexistentes	38,0	3	3	Doente
19	F	87	Espalhadas	39,0	6	6	Doente
34	M	67	Uniformes	38,4	2	2	Saudável

Alta correlação supervaloriza um dado aspecto dos dados e tornam mais lento o processo de indução do modelo.

(III) Redundância com os rótulos



Alta correlação torna a *feature* determinante para a predição do rótulo.

Dados com ruídos

Dados com ruídos

- Aparentemente não pertencem à distribuição que gerou os dados analisados.
 - variância ou erro aleatório no valor medido para uma *feature*.
 - podem levar a um **super-ajuste** do modelo.
- ***Outliers***: valores além dos limites aceitáveis ou são bastante diferentes dos demais valores observados para a mesma *feature*

Idade	Sexo	Peso
28	M	79
18	F	300
49	M	92
18	M	43
21	F	52
22	F	72
19	F	87
34	M	67



Dados com ruídos: algumas soluções

- **Técnicas de encestamento:** suavizar o valor de uma feature
 1. Valores p/ uma feature são ordenados;
 2. Divididos em cestas (faixas) com mesmo número de valores;
 3. Valores numa mesma cesta são substituídos pela média ou mediana da cesta
- **Técnicas baseadas em agrupamento:** valores de atributos que formarem clusters isolados são definidos como ruídos
- **Técnicas baseadas em distância:** verificam a que classe pertencem as instâncias mais próximas de cada outra instância x. Se as instâncias mais próximas pertencem a outra classe, são boas as chances da instância x apresentar ruído

Dados desbalanceados

Dados desbalanceados

- Em vários conjuntos de dados reais, o número de instâncias de cada classe varia bastante
 - Ex: dados dos pacientes de um hospital: 80% estão doentes e 20% saudáveis
 - Ex: clientes de um banco: 95% com saldo positivo no fim do mês e 5% com saldo negativo
- Modelos com dados desbalanceados tendem a favorecer a classificação de novos dados na classe majoritária
- Para ser aceitável, a **acurácia preditiva de um classificador deve ser maior que a acurácia obtida ao se atribuir toda nova instância à classe majoritária**

Técnicas para balanceamento

- Redefinir o tamanho do conjunto de dados
 - Excluir instâncias da classe majoritária
 - Criar instâncias da classe minoritária
 - Q: Problemas?
- Utilizar diferentes custos de classificação para as diferentes classes
 - Dificuldade de incorporar este conceito em modelos de aprendizado

Dados categóricos

Leitura de dataset

```
1 import pandas as pd  
2 df = pd.DataFrame([  
3     ['green', 'M', 10.1, 'class1'],  
4     ['red', 'L', 13.5, 'class2'],  
5     ['blue', 'XL', 15.3, 'class1']])  
6  
7 df.columns = ['color', 'size', 'price', 'classlabel']  
8 df
```

Nominais

Ordinais

Numéricos

	color	size	price	class label
0	green	M	10.1	class1
1	red	L	13.5	class2
2	blue	XL	15.3	class1

(I) Mapeando *features* ordinais para inteiros

```
1 size_mapping = {  
2     'XL': 3,  
3     'L': 2,  
4     'M': 1}  
5  
6 df['size'] = df['size'].map(size_mapping)  
7 df
```

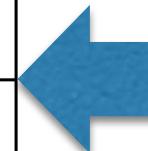


	color	size	price	class label
0	green	1	10.1	class1
1	red	2	13.5	class2
2	blue	3	15.3	class1

(II) Mapeando rótulos de classe para inteiros

```
1 import numpy as np  
2  
3 class_mapping = {label:idx for idx,label in enumerate(np.unique(df['classlabel']))}  
4 df['classlabel'] = df['classlabel'].map(class_mapping)
```

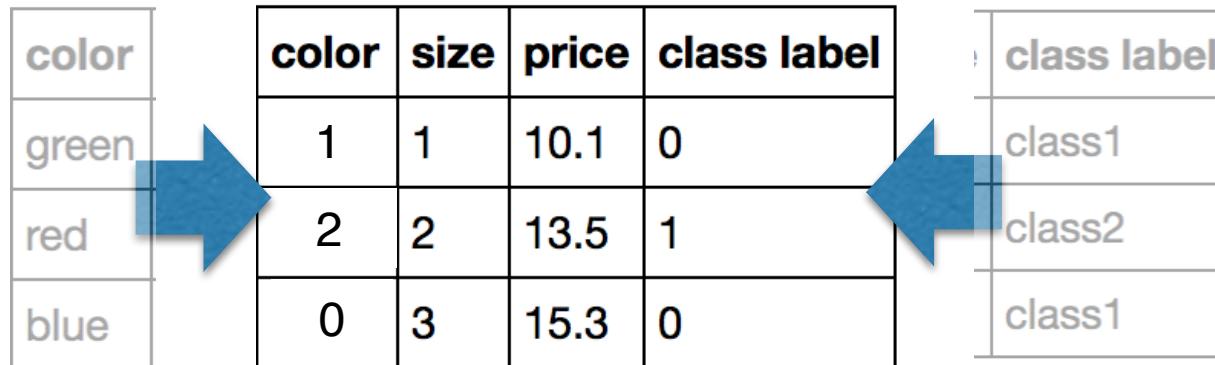
	color	size	price	class label
0	green	1	10.1	0
1	red	2	13.5	1
2	blue	3	15.3	0



class label
class1
class2
class1

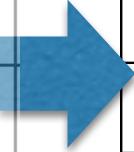
(III) Mapeando *features* nominais para inteiros

Q: Podemos trabalhar de forma semelhante aos rótulos?



(III) Mapeando *features* nominais para inteiros

Q: Solução?



color	size	price	class label
green	?	1	10.1
red	?	2	13.5
blue	?	3	15.3

(III) Mapeando *features* nominais para inteiros

```
1 pd.get_dummies(df[['price', 'color', 'size']])
```

Codificação **one-hot**



	price	size	color_blue	color_green	color_red
0	10.1	1	0	1	0
1	13.5	2	0	0	1
2	15.3	3	1	0	0

Reescala de dados

(I) Normalização .: [0, 1]

Para cada *feature*,

$$x_{norm}^{(i)} = \frac{x^{(i)} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$$

(II) Estandardização

Features: média = 0 (zero) e desvio = 1 (um) (distribuição normal)

→ facilita aprendizado de pesos

→ mantém informação sobre *outliers*, fazendo modelo menos sensível aos mesmos.

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

Divulgação científica



Hendrik Macedo

Escreve sobre Inteligência Artificial no Saense.

<http://www.saense.com.br/autores/artigos-publicados-por-hendrik-macedo/>