Fundamentos Matemáticos e Computacionais de Machine Learning

Especialização em Machine Learning e Big Data

UEL

Profa. Dra. Juliana Felix

jufelix16@uel.br

Revisão

Dataset

Dataset é um conjunto de dados que combina amostra com

- Valores ou variáveis de **entrada** (features, características) e
- Valores de **saída** (outcome, labels) utilizados no aprendizado supervisionado

Dataset

Exemplo:

Preço de lotes na 'Terra tão tão distante'	
Tamanho do lote (em m^2) - X	**Preço do lote (R$) - Y**
2104	399.900
1600	329.900
2400	369.000
...	...

Total amostras = 47

Feature (característica)

Outcome (saída)

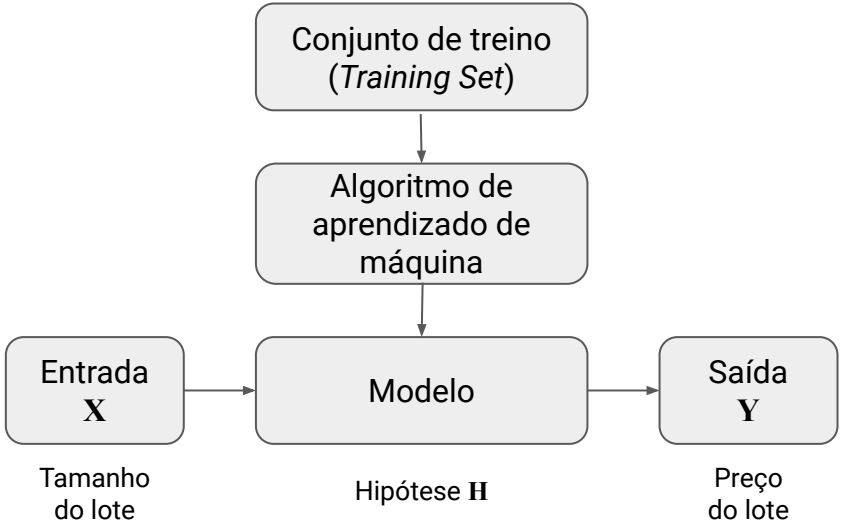
Notação

Podemos pensar no problema anterior como um problema que tem:

- Um total de $\mathbf{m}$ amostras/samples ($\mathbf{m}$ = 47)
- Cada amostra tem 1 única feature/característica (tamanho do lote)
  - Costumamos representar uma variável de entrada por $\mathbf{x}$
- Para cada amostra, temos uma única saída (preço do lote).
  - Costumamos representar uma variável de saída por $\mathbf{y}$
- Cada amostra pode ser representada por um par, ou tupla $\mathbf{(x,y)}$
  - Uma tupla $\mathbf{(x^i, y^i)}$ representa a i-ésima amostra do problema, com $\mathbf{1 \leq i \leq m}$

Processo básico de Machine Learning

A base de qualquer processo de machine learning consiste em mapear um dado de entrada $\mathbf{X}$ em um dado de saída $\mathbf{Y}$.



Problemas e Soluções

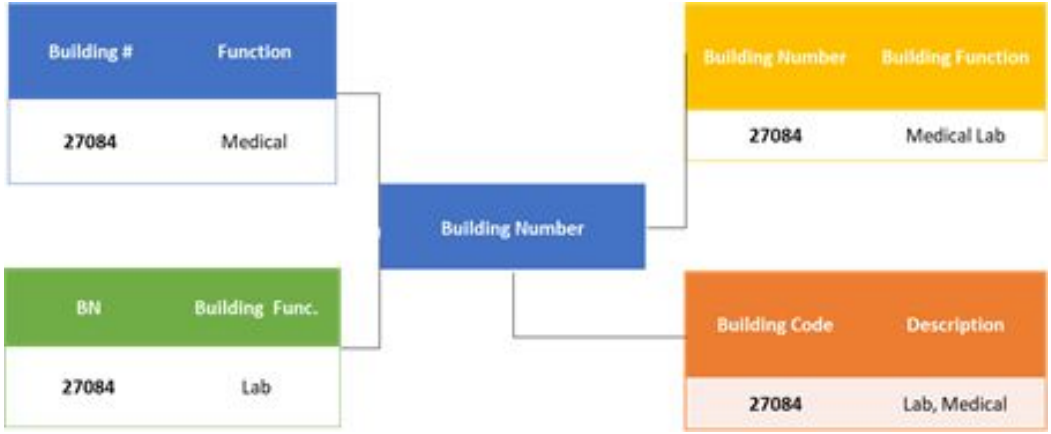
Dataset

Quando vamos trabalhar com um novo dataset, podemos enfrentar uma série de problemas:

- Dados inconsistentes
- Dados faltantes
- Dados duplicados
- Dados irrelevantes
- Dados "sujos"
- Dados ruidosos

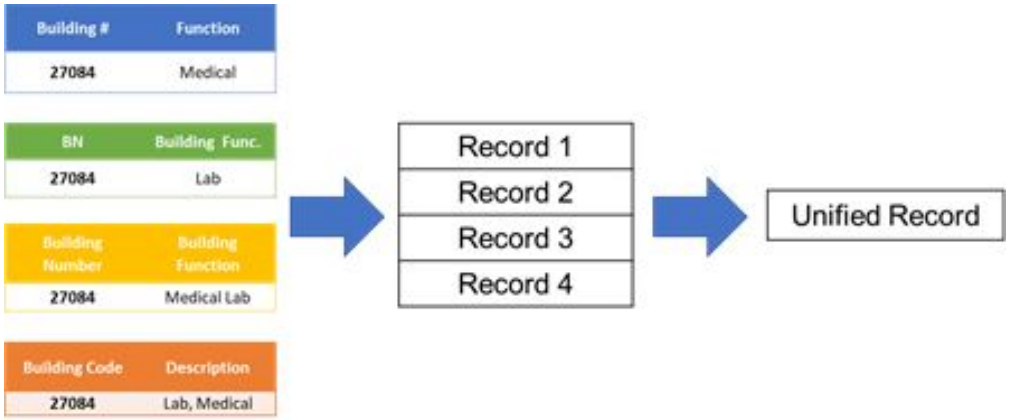
Dados Inconsistentes

Muitas informações no seu dataset podem estar dispostas de modo inconsistente, ou mesmo haver vários registros que fazem referência à uma mesma informação, cabendo ao programador observar este detalhe e fazer os ajustes necessários.



Dados Inconsistentes

Uma solução para isso é a unificação dos dados



Dados Inconsistentes

Em alguns casos, unificar os dados pode não ser tão bom...

O que fazer?

Financial

Employee	Salary
John	1000

Employee → Salary

Human Resources

Employee	Salary
John	2000
Mary	3000

Employee → Salary

Target Database

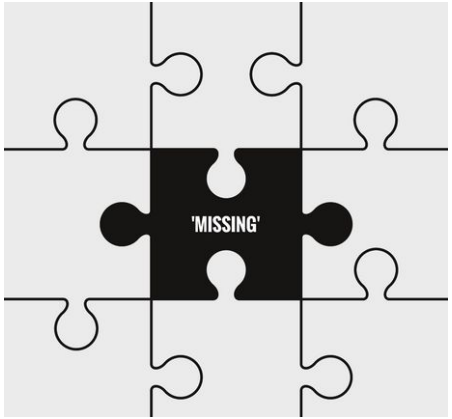
Employee	Salary
John	1000
John	2000
Mary	3000

Employee → Salary

Dados faltantes

Um dataset com dados faltantes é um problema, visto que faltam informações que poderiam ser importantes para passar para o modelo.

Ex.: dados categóricos ou dados numéricos podem estar faltando no seu conjunto de dados.



Dados faltantes

A solução mais simples (e ingênua) para lidar com "missing data" é:

- deletar a amostra com dados faltantes, ou
- desconsiderar a "feature" que possui dados faltantes.

Quais os problemas com essa abordagem?

Dados faltantes do tipo "categóricos"

Uma das melhores formas de lidar com dados faltantes do tipo "categóricos"é simplesmente rotulá-los como "missing"!

- Você estará, basicamente, criando uma nova classe para dados deste tipo.
- Isso dirá ao algoritmo que há informações não disponíveis.
- E isso também resolverá o requerimento técnico de que não devem haver valores faltantes.

Dados faltantes do tipo "numéricos"

Para dados faltantes do tipo "numéricos", devemos:

- **sinalizar** o dado faltante como "missing" utilizando um indicador próprio para isto e
- **preencher** o valor faltante com o valor 0 (ou outro valor desejado) para cumprir o requisito técnico de não haver dados faltantes

Dados faltantes do tipo "numéricos"

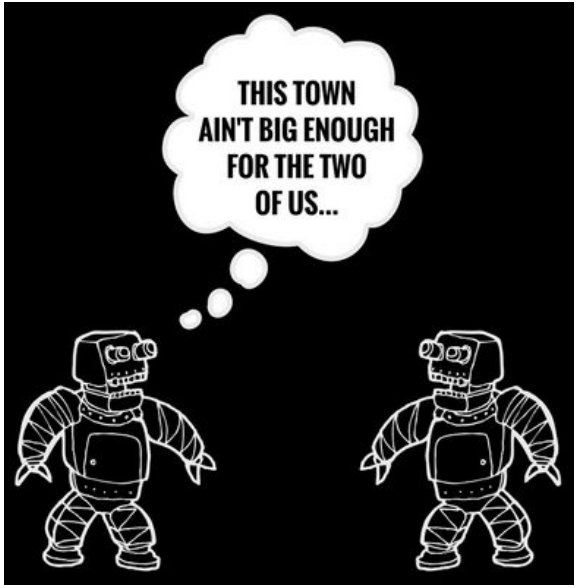
Ao utilizar esta técnica de "sinalizar e preencher", estamos, essencialmente

- permitindo que o algoritmo estime uma constante ótima para estes valores ao invés de simplesmente preencher o valor manualmente (com a média do grupo, por exemplo)

Dados duplicados

Ao inspecionar os dados, podemos perceber que há dados ou entradas duplicadas, o que pode fazer com que o modelo se ajuste de forma tendenciosa àquele dado ou saída.

- Uma estratégia comum é simplesmente deletar a entrada duplicada.



Dados duplicados

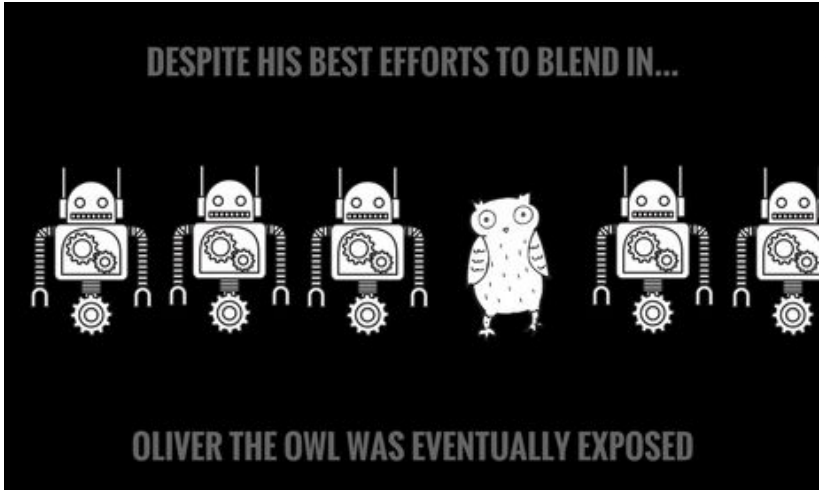
id	first_name	last_name	email
1	Carine	Schmitt	carine.schmitt@verizon.net
4	Janine	Labrune	janine.labrune@aol.com
6	Janine	Labrune	janine.labrune@aol.com
2	Jean	King	jean.king@me.com
12	Jean	King	jean.king@me.com
5	Jonas	Bergulfsen	jonas.bergulfsen@mac.com
10	Julie	Murphy	julie.murphy@yahoo.com
11	Kwai	Lee	kwai.lee@google.com
3	Peter	Ferguson	peter.ferguson@google.com
9	Roland	Keitel	roland.keitel@yahoo.com
14	Roland	Keitel	roland.keitel@yahoo.com
7	Susan	Nelson	susan.nelson@comcast.net
13	Susan	Nelson	susan.nelson@comcast.net
8	Zbyszek	Piestrzeniewicz	zbyszek.piestrzeniewicz@att.net

S.N°	First Name	Last Name	Title	Company
1	Mary	Sue	Senior Marketing Manager	ABC Ltd.
2	Janet	Martin	Marketing Executive	ABC Ltd.
3	Bryan	Oscar	SEO Manager	ABC Ltd.
4	Jude	Taylor	Marketing Manager	ABC Ltd.
5	Mary S	Sue	Senior Marketing Manager	ABC Ltd.

Dados irrelevantes

Dados irrelevantes são aqueles que não necessariamente se encaixam num problema específico que deseja-se resolver.

- Se estivermos construindo um modelo para predizer valores de casas na região de Londrina-PR, não queremos que os dados que relacionam preços de casas em Goiânia façam parte do modelo.



Dados "sujos"

Alguns dados podem conter informações dispostas de forma "bagunçada" ou que não seguem um padrão.

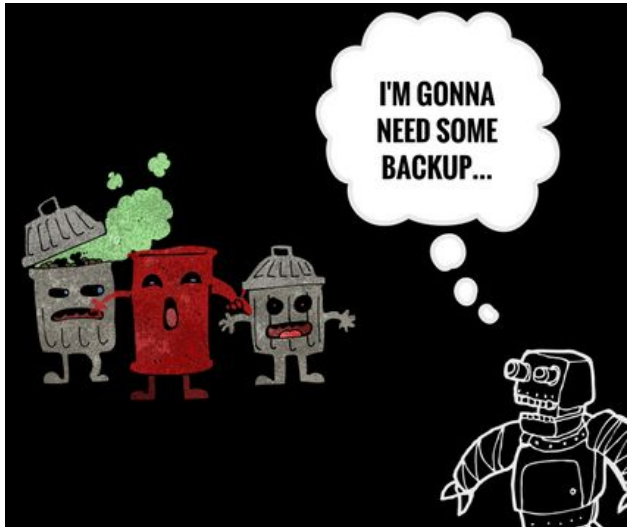
Por exemplo:

- Uma string pode ter caracteres de espaço "extras", ou caracteres "estranhos"
- Números podem estar dispostos em formatos diferentes ao longo do dataset
    - Altura pode estar representada em metros (1.75) e em cm (175)

Dados "sujos"

Devemos inspecionar os dados buscando resolver esses problemas, quando possível, ou deletá-los, quando for o caso.

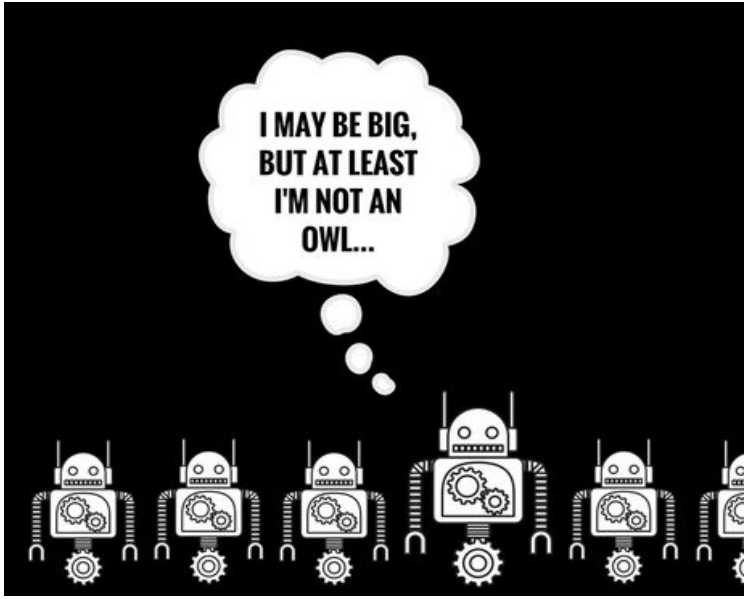
Exemplo: Examples of Dirty Data



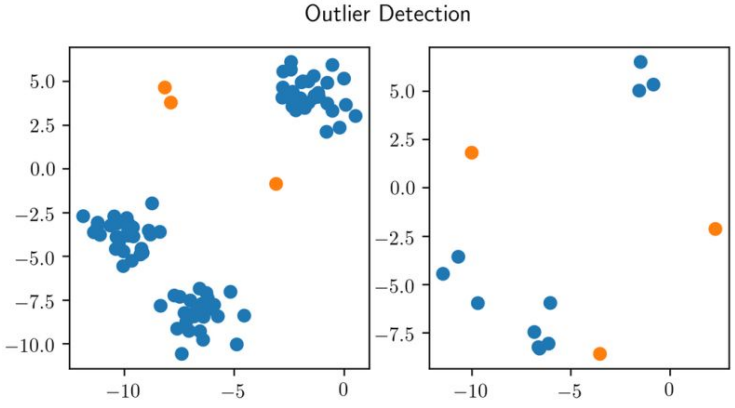
Dados ruidosos

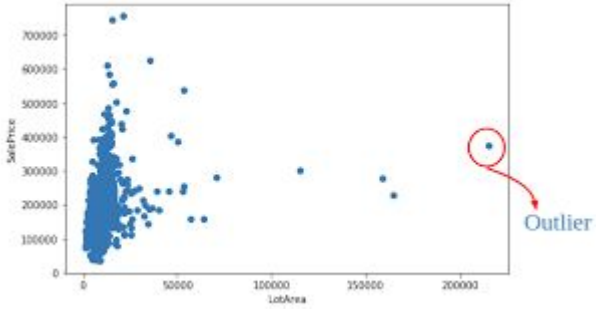
Alguns dados podem conter valores fora do limite esperado para certas características, sendo conhecidos como "**outliers**".

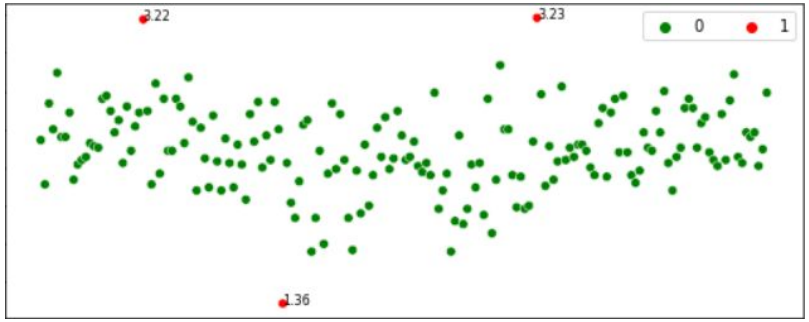
Por mais que essas informações possam representar dados reais, ou serem corretas, estes valores fora do padrão podem levar a falsas tendências no seu modelo.

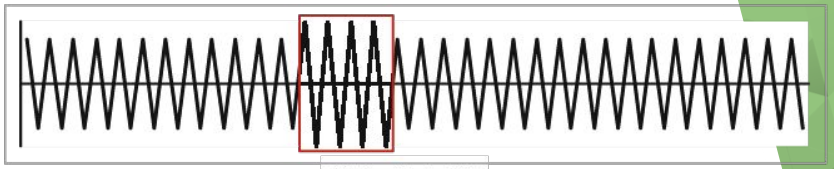


Outliers

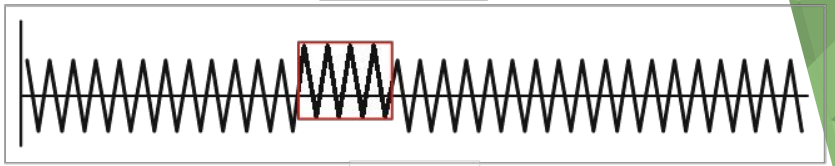




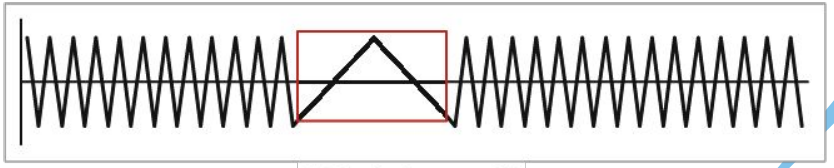




(a) Amplitude shift



(b) Level shift



(c) Direction frequency shift

Boas práticas

Transformação dos dados

Transformação de dados

Além dos diversos problemas que podemos nos deparar ao lidar com um novo dataset, também devemos nos preocupar em **analisar os vários dados disponíveis** no dataset e **garantir que estejam prontos** para alimentar um novo modelo de aprendizado de máquina.

Transformação de dados

Para garantir que os dados estejam prontos para alimentar um novo modelo, pode ser necessário realizar uma

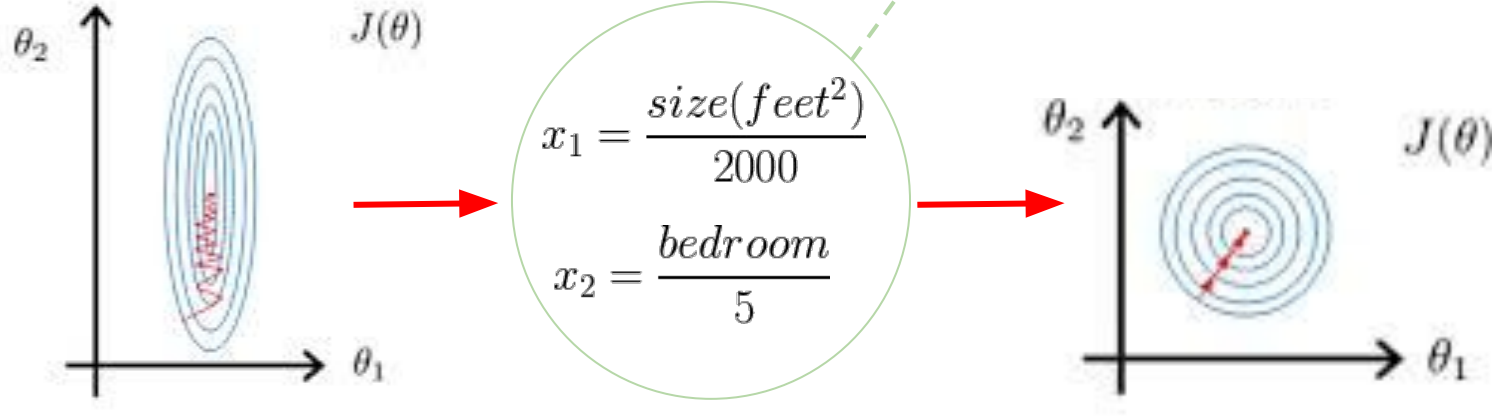
- Escala
- Normalização
- Transformação de dados
- ou Mapeamento de dados

Normalização dos dados

É importante que as features estejam numa mesma "escala".

- x_1 área (0–2000 m^2)
- x_2 número de quartos (1–5)

$$x'_j = \frac{x_j}{max(x_j)}, 0 \leq x_j \leq 1$$



Normalização dos dados

Exemplo: Fazer a normalização/escala utilizando o valor máximo

Preço max

Área max

Área (m^2)	Preço (R$1000)
2104	460
1416	232
1534	315
852	178

Área (m^2)	Preço (R$1000)
2104/2104	460/460
1416/2104	232/460
1534/2104	315/460
852/2104	178/460

Área (m^2)	Preço (R$1000)
1	1
≅0.673	≅0.504
≅0.729	≅0.685
≅0.405	≅0.387

Normalização dos dados

Ao normalizar os dados, seu modelo será construído baseado nos dados normalizados e, consequentemente, ...

- Os valores de Theta encontrados não refletirão os dados atuais.

Para utilizar o seu modelo para estimar valores, como na regressão linear, será necessário:

- escalar/normalizar as variáveis de entrada
- fazer o processo inverso para obter a saída equivalente

Normalização dos dados

Exemplo: Fazer a normalização/escala utilizando o valor máximo

Área max

Área (m^2)	Preço (R$1000)
2104	?
1416	?
1534	?
852	?

Área (m^2)	Preço (R$1000)
2104/2104	?
1416/2104	?
1534/2104	?
852/2104	?

Área (m^2)	Preço (R$1000)
1	?
≅0.673	?
≅0.729	?
≅0.405	?

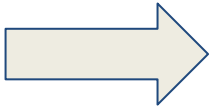
Escala

Normalização dos dados

Exemplo: Fazer a normalização/escala utilizando o valor máximo

Área (m^2)	Preço (R$1000)
2104	?
1416	?
1534	?
852	?

PREDIÇÃO



Área (m^2)	Preço (R$1000)
1	1
≅0.673	0.504
≅0.729	0.685
≅0.405	0.387

Normalização dos dados

Exemplo: Fazer a normalização/escala utilizando o valor máximo

Área * 2104

Preço * 460

Área (m^2)	Preço (R$1000)
1	1
≅0.673	0.504
≅0.729	0.685
≅0.405	0.387

Área (m^2)	Preço (R$1000)
2104	460
1416	232
1534	315
852	178

Processo inverso

Normalização dos dados

Outras formas de normalizar os dados:

- Mean normalization

$$x'_j = \frac{x_j - \mu(x_j))}{max(x_j)}, -0.5 \leq x_j \leq +0.5$$

μ - média de x_j

34,03	0,20
20,48	-0,18
14,67	-0,34
29,75	0,08
30,50	0,10
36,20	0,26
21,52	-0,15
33,47	0,18
16,23	-0,29
32,04	0,14

Necessário para realizar a de-normalização também

Média	26,89
Max	36,20

Normalização dos dados

Outras formas de normalizar os dados:

- Mean-range normalization

$$x'_j = \frac{x_j - \mu(x_j))}{max(x_j) - min(x_j)}$$

μ - média de x_j

34,03	0,33
20,48	-0,30
14,67	-0,57
29,75	0,13
30,50	0,17
36,20	0,43
21,52	-0,25
33,47	0,31
16,23	-0,50
32,04	0,24

Necessário para realizar a de-normalização também

Média	26,89
Max	36,20
Min	14,67

Normalização dos dados

Outras formas de normalizar os dados:

- Z-score normalization

$$x'_j = \frac{x_j - \mu(x_j))}{s(x_j)}$$

μ - média de x_j

s - desvio padrão de x_j

- Dados transformados terão
  - Média 0, e
  - Desvio Padrão 1

34,03	0,90
20,48	-0,81
14,67	-1,55
29,75	0,36
30,50	0,46
36,20	1,18
21,52	-0,68
33,47	0,83
16,23	-1,35
32,04	0,65
Média 26,89	Média 0,00
Desvio Padrão 7,90	Desvio Padrão 1,00

Necessário para realizar a de-normalização também

Dataset splitting

Aprendizado Supervisionado

Por que treinamos um modelo de Machine Learning?

- Para aproximar uma função (regressão)
- Para determinar se uma amostra pertence a um grupo (classificação)

Preparando um Experimento

- Como sabemos se o modelo criado está funcionando?
- Se o erro do modelo for 0% (ou, equivalentemente, 100% de acerto) ao fim do treinamento, isso significa que o modelo está preparado para trabalhar com novos dados?

Exemplo

Se você tiver dados históricos de clientes que fizeram empréstimo de um banco...

Exemplo

E você tiver treinado um modelo para classificar se um cliente é um "bom pagador" ou não...

Exemplo

É preciso emprestar o dinheiro para checar se o modelo criado funcionado?

E se não estiver funcionando como esperado?

Preparando um Experimento

Se você tiver utilizado todo o dataset, serão necessários novos dados para checar se o modelo está funcionando como deveria.

Alguns problemas:

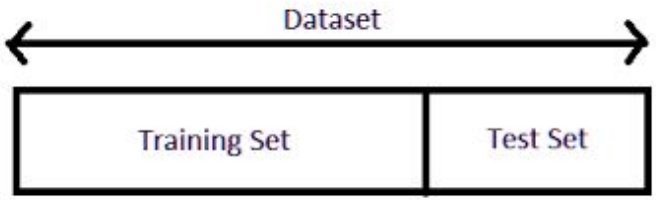
- Quanto tempo levará para obter novos dados?
- Quais as consequências se o modelo estiver falho?

Preparando um Experimento

O que fazer se não pudermos aguardar até que novos dados sejam coletados?

- Não utilizar todo o dataset para a fase de treinamento.

Treino/Teste



Conjunto de treinamento

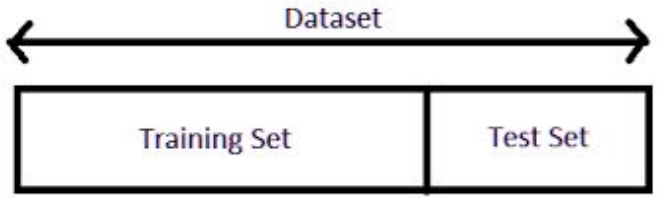
- É um subconjunto do dataset separado especialmente para ser utilizado na fase de treinamento do modelo.

Conjunto de teste

- É um subconjunto do dataset separado especialmente para ser utilizado na fase de teste do modelo.

Treino/Teste

- Como dividir o conjunto de dados em conjuntos de treinamento e de teste?
- Quão grande deve ser cada conjunto?



Treino/Teste

A abordagem mais simples consiste em dividir o conjunto utilizando uma **porcentagem** dos dados para treino e teste.

Os valores mais comuns são:

- 90 / 10 para treino / teste
- 80 / 20 para treino / teste
- 75 / 25 para treino / teste
- 70 / 30 para treino / teste

Em geral, o conjunto de treinamento é maior que o conjunto de teste.

Exemplo

Considerando o nosso exemplo de preço de casa/lote que contém 47 amostras:

Treino	Teste	Total
90.00%	10.00%	100.00%
42.3	4.7	47
80.00%	20.00%	100.00%
37.6	9.4	47
75.00%	25.00%	100.00%
35.25	11.75	47
60.00%	40.00%	100.00%
28.2	18.8	47

Exemplo

Ao ajustar os valores...

Treino	Teste	Total
90.00%	10.00%	100.00%
42.3	4.7	47
80.00%	20.00%	100.00%
37.6	9.4	47
75.00%	25.00%	100.00%
35.25	11.75	47
60.00%	40.00%	100.00%
28.2	18.8	47

Treino	Teste	Total
89.36%	10.64%	100.00%
↓ 42	↑ 5	47
80.85%	19.15%	100.00%
↑ 38	↓ 9	47
78.72%	21.28%	100.00%
↓ 37	↑ 10	47
74.47%	25.53%	100.00%
↓ 35	↑ 12	47
76.60%	23.40%	100.00%
↑ 36	↓ 11	47
61.70%	38.30%	100.00%
↑ 29	↓ 18	47

Treino/Teste

A biblioteca sklearn provê uma função para realizar a separação (split) dos dados em treino/teste:

- sklearn.model_selection.train_test_split — scikit-learn 1.2.2 documentation

Outras formas de divisão de dados em treino e teste serão vistas na disciplina de Machine Learning.

Exercício

Faça um código em Python que leia o arquivo *price-house.txt* e realize o split em conjuntos de treino/teste com as seguintes proporções:

- 90/10;
- 80/20;
- 75/25;
- 70/30;
- 60/40;

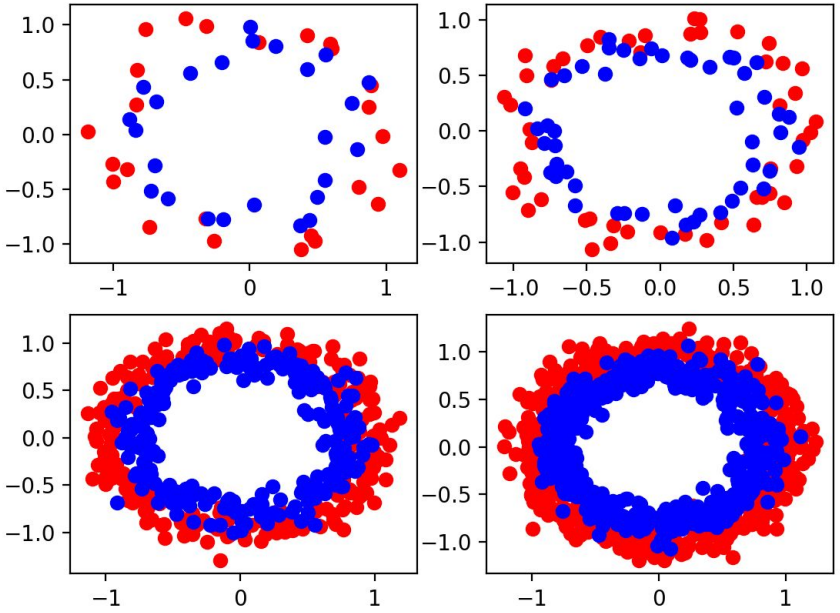
Plote um gráfico de scatter plot colorindo o conjunto de treinamento e o de teste em cada divisão.

Você observou algum padrão nas figuras geradas?

Principais Desafios

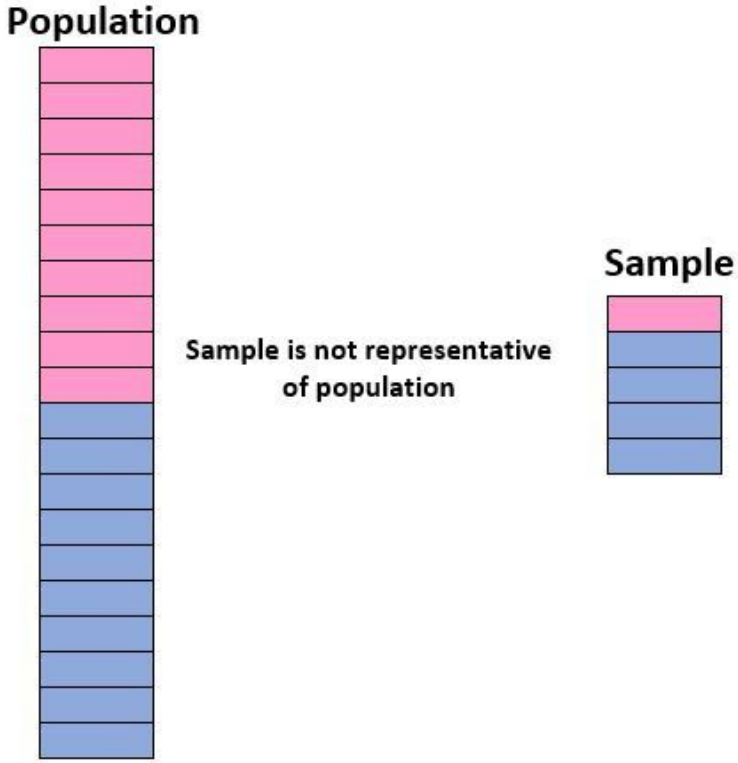
Principais Desafios

- Dados de treinamento insuficientes



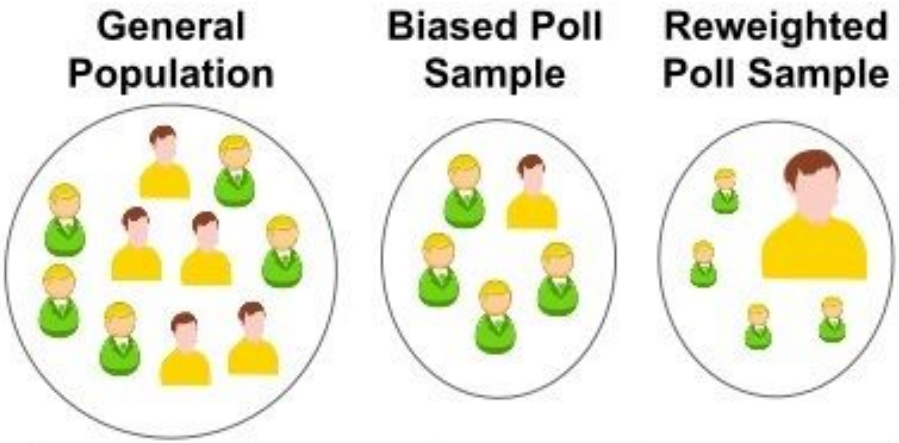
Principais Desafios

- Conjunto de treinamento não-representativo



Principais Desafios

- Conjunto de treinamento não-representativo



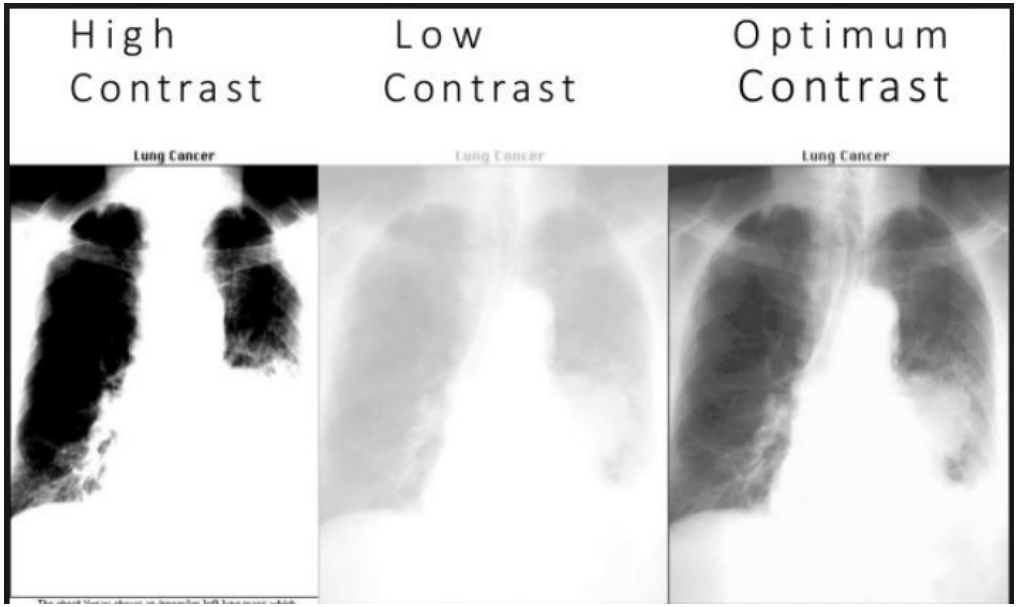
Movie Name	Population Rating	Biased Poll Rating	Reweighted Poll Rating
	3.57	2.56	3.67

Principais Desafios

- Dados de baixa qualidade

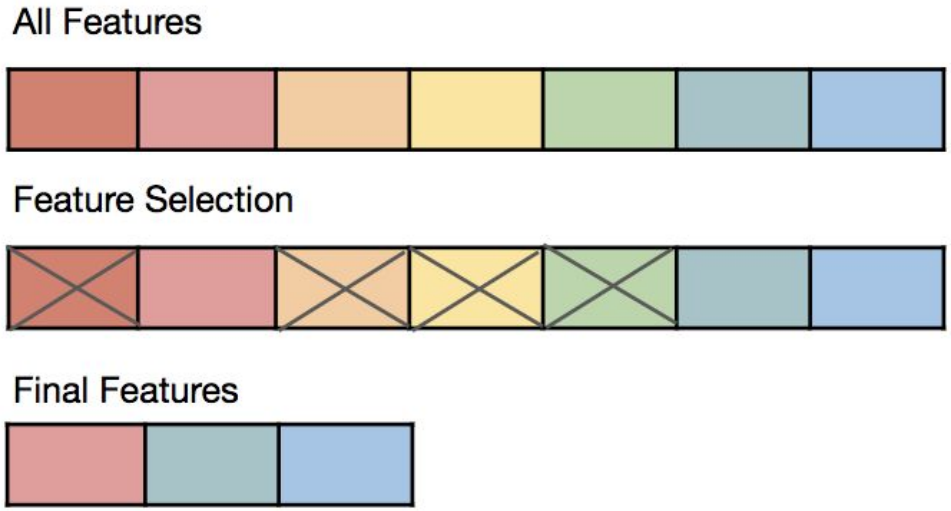
Principais Desafios

- Dados de baixa qualidade



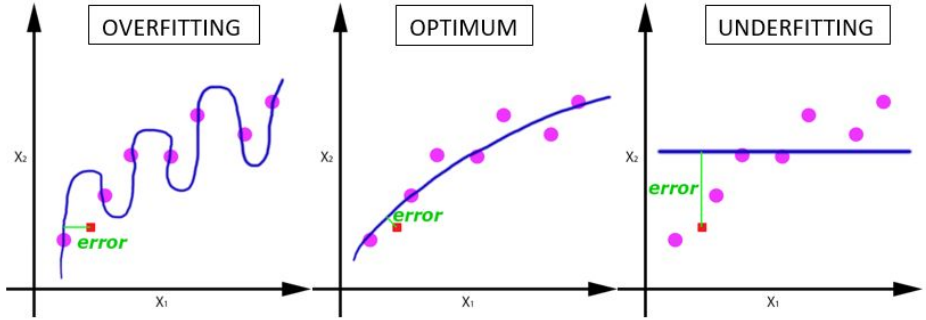
Principais Desafios

- Features (características) irrelevantes



Principais Desafios

- Dados de treinamento insuficientes
- Conjunto de treinamento não-representativo
- Dados de baixa qualidade
- *Features* irrelevantes
- *Overfitting* do conjunto de treinamento
- *Underfitting* do conjunto de treinamento



Lista 4 - Bibliotecas e Dataset, entrega até sexta-feira, dia 26/05