



Descoberta do Conhecimento





Descoberta do Conhecimento

Prof. Cleilton Lima Rocha

Universidade 7 de Setembro
Fortaleza - CE, Brasil



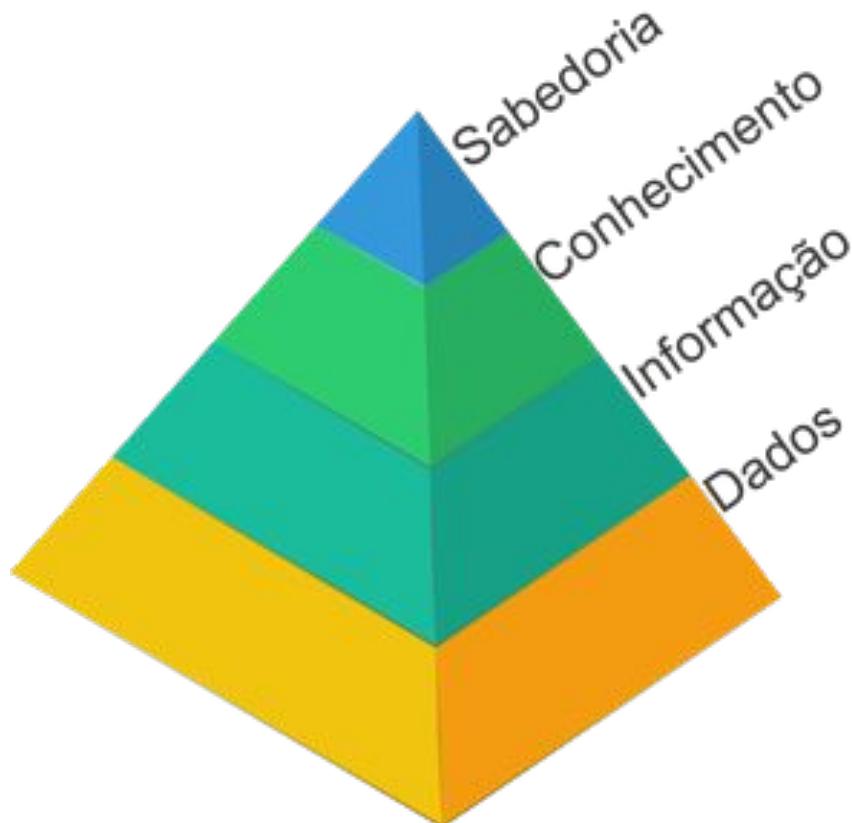


Agenda

- ◊ Introdução ao Processo de Descoberta de Conhecimento e Data Science
 - ◊ Feature engineering:
 - Pré-processamento de dados
 - Modelagem dos dados
 - Seleção de Features ...
 - ◊ Modelos de aprendizagem supervisionada e não supervisionada
 - ◊ Análise do *bias variance threshold*
 - ◊ Introdução a Deep Learning
 - ◊ Projeto prático aplicado à Data Science.
- 



Processo de Descoberta de Conhecimento





“O KDD pode ser visto como o processo de descoberta de padrões e tendências por análise de grandes conjuntos de dados, tendo como principal etapa o processo de mineração, consistindo na execução prática de análise e de algoritmos específicos que, sob limitações de eficiência computacionais aceitáveis, produz uma relação particular de padrões a partir de dados FAYYAD et al (1996).”



“Informação é o resultado do processamento de dados num formato que tem significado para o usuário respectivo e que tem valor real ou potencial nas decisões presentes ou prospectivas DAVIS (1974).”

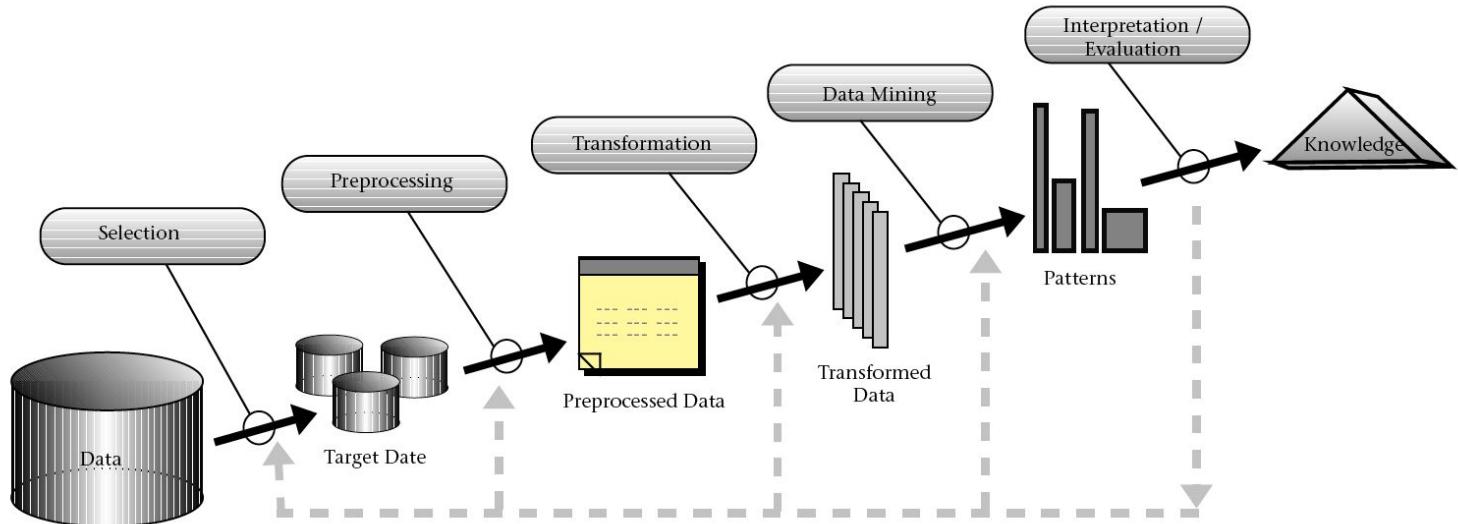


“Segundo DAVENPORT e PRUSAK (1998), a GC pode ser vista como uma série de ações gerenciais constantes e sistemáticas que facilitam os processos de criação, registro e compartilhamento do conhecimento nas organizações.”



“O conhecimento necessário para se decidir e/ou avaliar torna-se disponível por meio de informações SANCHES (1997).”

Fases do KDD





Data Mining e seus métodos

- ◊ Aprendizagem supervisionada
- ◊ Aprendizagem não supervisionada
- ◊ Modelos de regras de associação
- ◊ Modelos de relacionamento entre variáveis



ATD

Apoio à tomada de decisão





Interesse em Data Science

● data science
Termino de pesquisa

● big data
Termino de pesquisa

● machine learning
Termino de pesquisa

+ Adicionar comparação

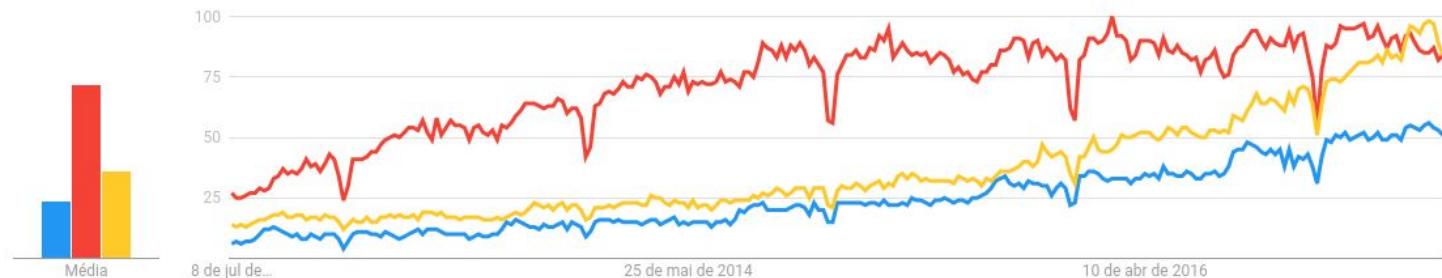
Todo o mundo ▾

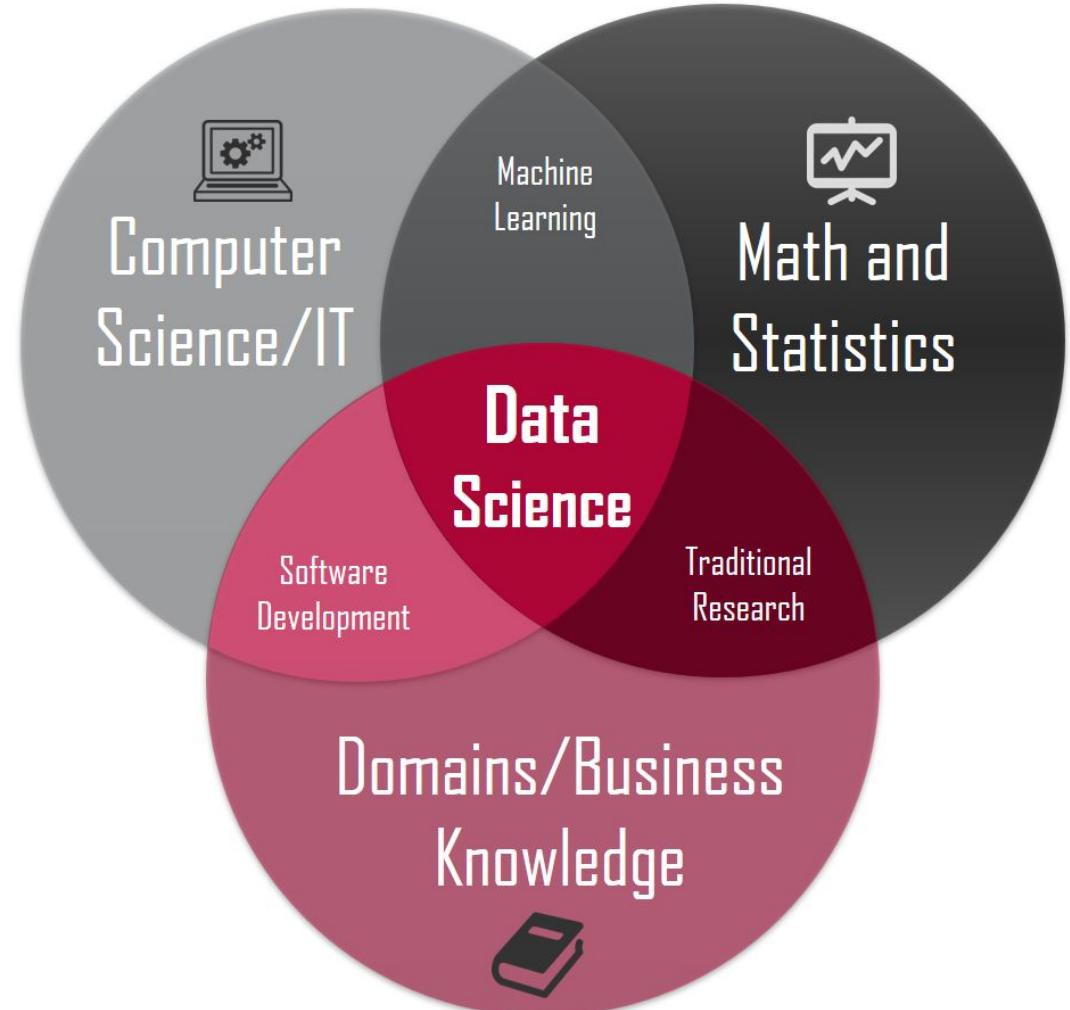
Nos últimos 5 anos ▾

Todas as categorias ▾

Pesquisa na Web ▾

Interesse ao longo do tempo ?

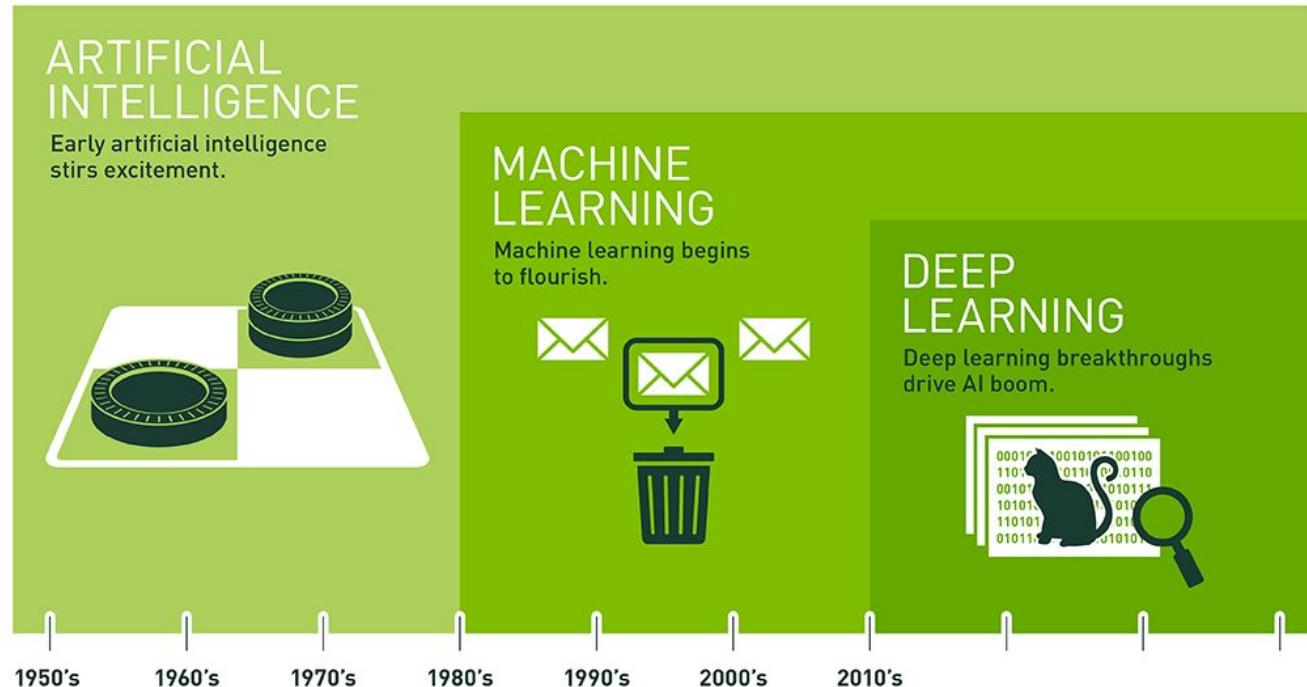




Data Science – uma ciência
interdisciplinar



Machine Learning Overview



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.



Data Science Overview

data **quantitative** **statistical** **inference** **models** **analysis** **deviation** **statistics**

research coefficient regression learning generalized linear bayesian probability modeling maximum research coefficient regression learning generalized linear bayesian probability modeling maximum

computing management expectation likelihood trend spatial visualization methods predictive normal parameter time function simulation workshops causal equation covariate duration variance distribution graphical standard

analytics expectation likelihood trend management spatial visualization methods predictive normal parameter time function simulation workshops causal equation covariate duration variance distribution graphical standard

expectation likelihood trend management spatial visualization methods predictive normal parameter time function simulation workshops causal equation covariate duration variance distribution graphical standard

likelihood trend management spatial visualization methods predictive normal parameter time function simulation workshops causal equation covariate duration variance distribution graphical standard

trend management spatial visualization methods predictive normal parameter time function simulation workshops causal equation covariate duration variance distribution graphical standard



Exemplos



Recommendation Systems



Inventory planning



Dynamic
pricing

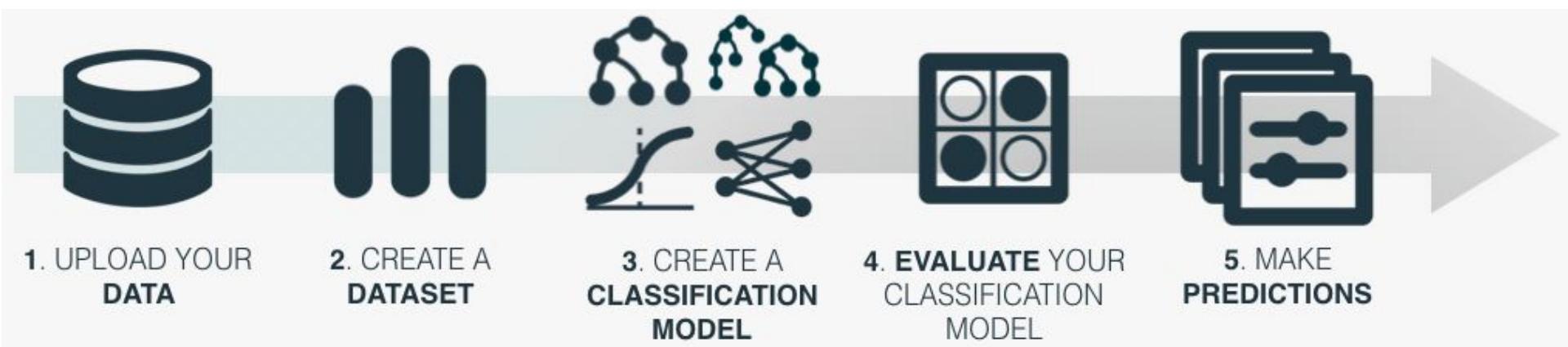


CRISP-DM





Visão geral de um modelo end-to-end



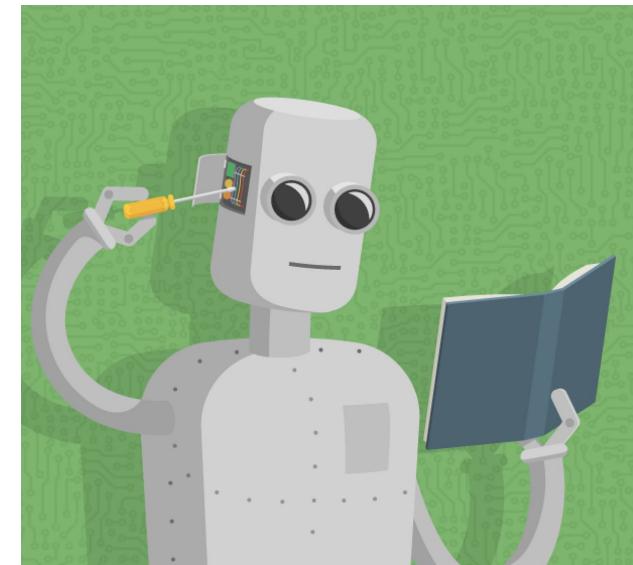
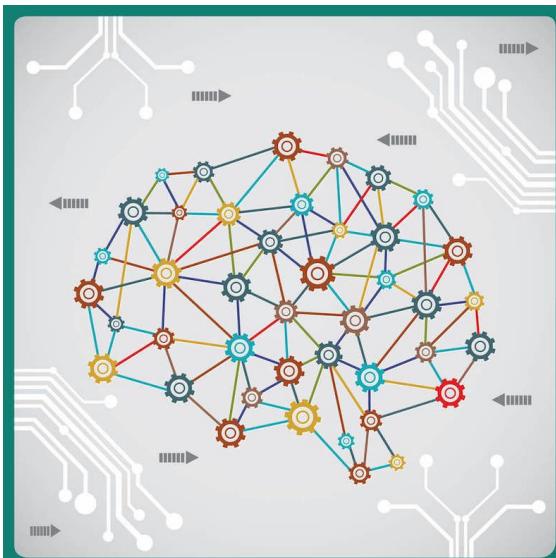


Conhecimentos desejáveis





Conhecimentos desejáveis

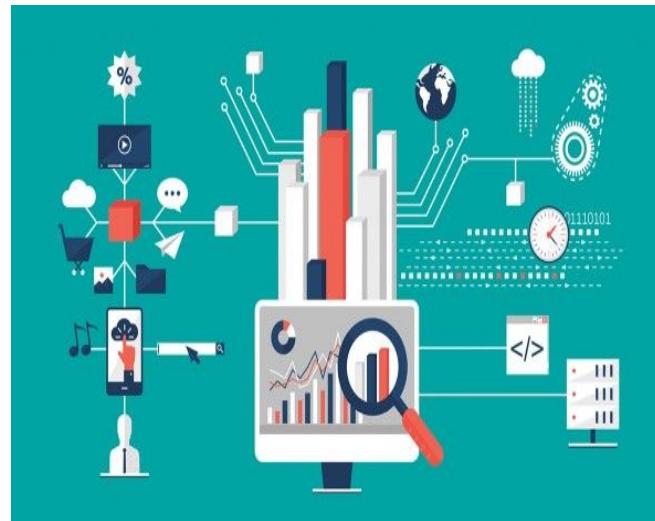




Conhecimentos desejáveis



Big Data



Processamento de stream e
séries temporais



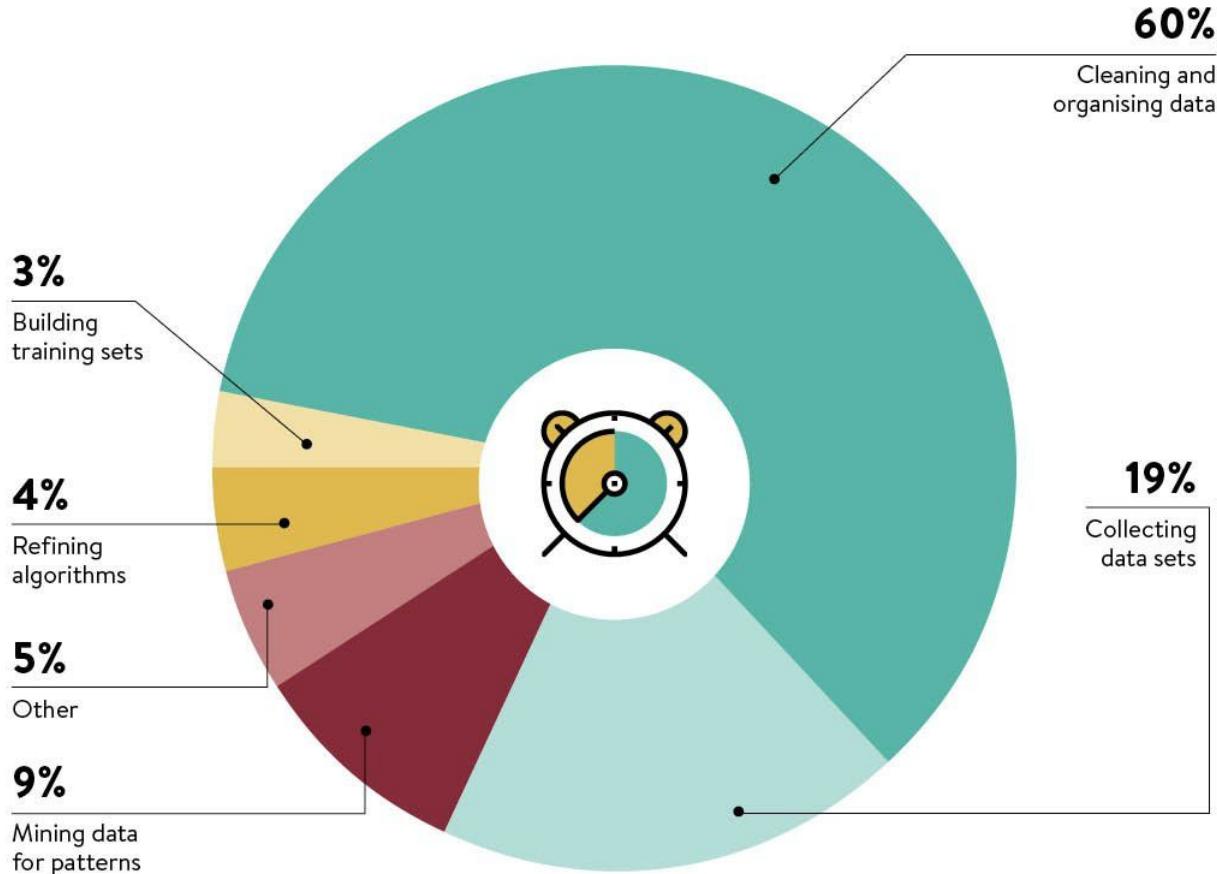
Processamento de Linguagem
Natural



Feature engineering



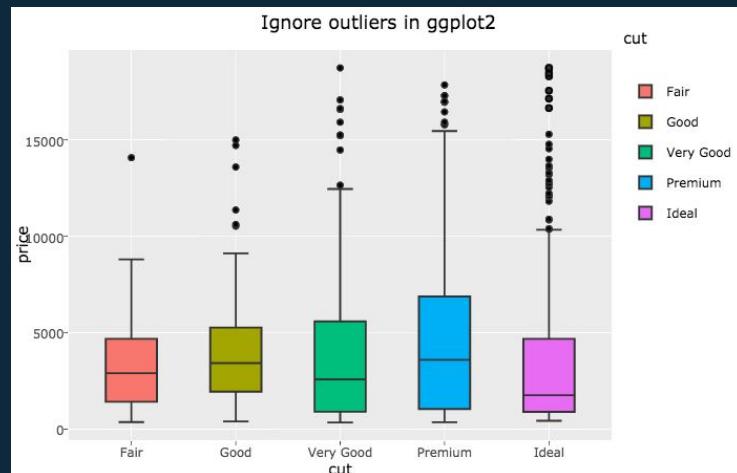
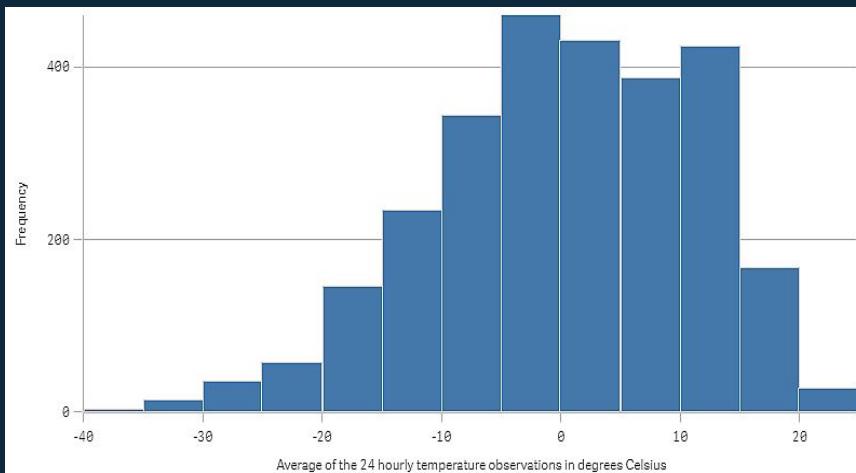
WHAT DATA SCIENTISTS SPEND THE MOST TIME DOING



Source: CrowdFlower 2016



Exploração do dado



Escalas de Dados

- ◊ **Nominal**: nessa escala os valores valores são não numéricos e não ordenados. Por exemplo, cor, marca de carro, etc.
- ◊ **Ordinal**: Nessa escala os valores não são numéricos, mas são **ordenados**. Uma amostra pode apresentar um valor comparativamente maior do que uma outra. Ex: Função no trabalho

Escalas de Dados

- ◊ **Intervalar**: escala onde valores são numéricos, existindo uma ordem entre os valores e uma diferença entre esses valores. O zero é relativo.
- ◊ **Proporcional**: nessa escala de valores numéricos, além da diferença, tem sentido calcular a proporção entre valores.

Os atributos podem ser:

◊ **Qualitativo:**

escalas
nominais ou
ordinais:

- Variáveis Discretas
- Binárias

◊ **Quantitativo:**

escalas
intervalar ou
proporcional

- Variáveis contínuas

Ausentes ou
inaplicáveis

Pré-processamento do dado



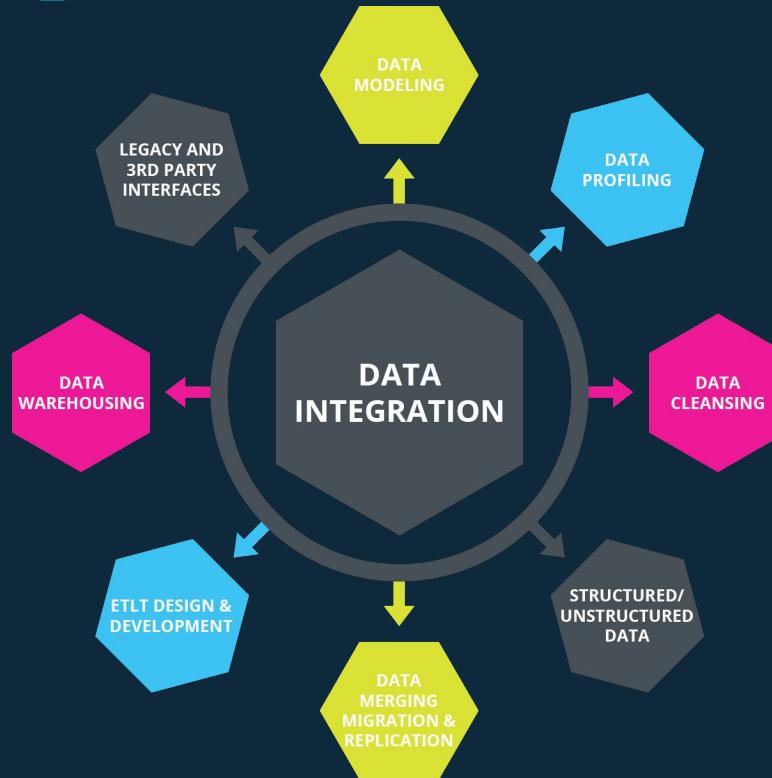


Coletar o dado

Coletar o dado

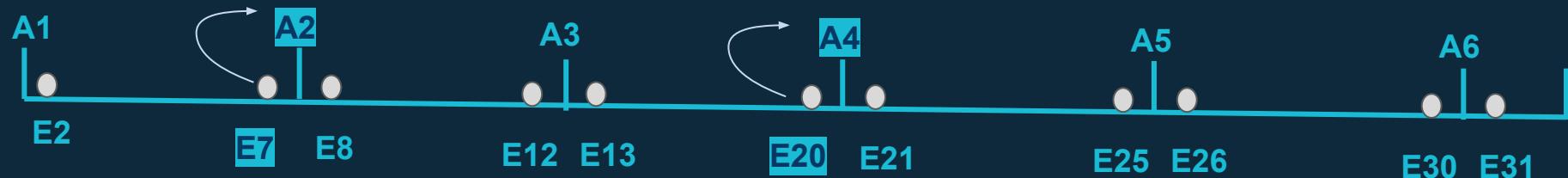
- ◊ Dados Públicos
- ◊ Dados no DBpedia
- ◊ Plataformas de ensino (e.g. Kaggle, UCI)
- ◊ Crowler
- ◊ REST API
- ◊ Acesso direto as fontes de dados
- ◊ Dado estruturado
- ◊ Dado não estruturado

Compreender e Integrar





Modelagem da amostra



Limpeza dos dados

- ◊ Preencher dados ausentes
 - Como preencher valores numéricos?
 - Como preencher valores nominais?
 - Aplicar ML
- ◊ Remover dados ausentes
 - Quando eliminar uma amostra?
 - Quando eliminar uma coluna?
- ◊ Identificar outlier
 - Qual a melhor fórmula?

Criação de features

- ◊ Aplicar Fórmula (e.g.: Faixa salarial)
- ◊ Combinação de duas features (e.g.: cargo e função)
- ◊ Valores proporcionais (e.g.: IMC)
- ◊ Opcional: Eliminar features originais

Agregação

- ◊ Combinar dois ou mais atributos (ou objetos) em apenas um atributo (ou objeto)
- ◊ Objetivo:
 - Reduzir o número de atributos ou amostras
 - Mudar escala (e.g: cidade em estado)
 - Possuir dados mais estáveis devido a menor variabilidade



“É necessário para obter os dados em uma forma apropriada para a aplicar data science com machine learning”

Transformação

- ◊ Label encoding
 - Sexo (F, M) → Sexo (F: 0), (M: 1)
- ◊ One Hot Encoding
 - Resulta em uma matriz esparsa

	Idade	Sexo_M	Sexo_F
Amastra _1	10	1	0
Amastra _2	30	0	1

Feature scaling

Normalização: o propósito é minimizar os problemas oriundos do uso de unidades e dispersões distintas entre as variáveis.

Normalização segundo a amplitude: unidades diferentes ou dispersões muito heterogêneas.

◆ Min e max norm:

$$Y = \frac{X - \text{min}}{(\text{Max} - \text{Min})}$$

◆ Média norma.:

$$Y = \frac{X - \text{media}}{(\text{Max} - \text{Min})}$$

◆ Standardization

$$Y = \frac{X - \text{media}}{\text{std}}$$

Feature scaling

Normalização distribucional: é interessante nas situações em que há distorção nos valores aberrantes, obtenção de simetria etc. Por exemplo: salário dos brasileiros

Exemplo mais comum:
◊ Log X
Salários (1000, 10000)



Pré-processamento do dado

Merging



MERGE



Seleção de features



Importância

- ◊ Otimizar modelo
- ◊ Facilita a interpretação
- ◊ Obter *insights*
- ◊ Reduzir o overfitting



Filter method



- ◊ É independente do modelo de aprendizagem
 - ◊ Pode ser feito com base no conhecimento do negócio
- Exemplos:
- ◊ Seleção manual
 - ◊ Correlação de Pearson
 - ◊ Chi Square

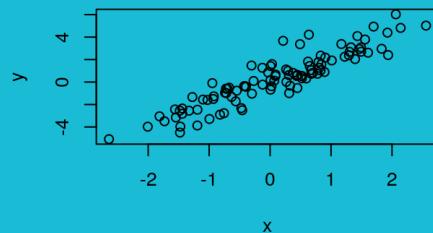
Filter method

Feature/Response	Contínua	Categórica
Contínua	Correlação de Pearson	LDA
Categórica	Anova	Chi-Square

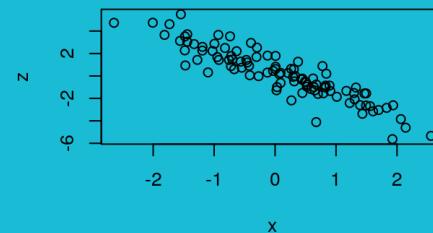
O que fazer com variáveis que são fortemente correlacionadas?

Correlação de pearson

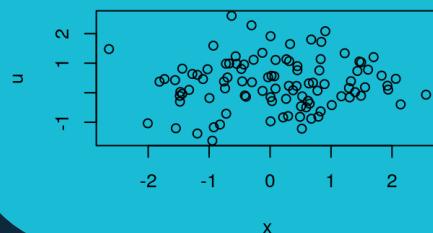
Relação linear positiva



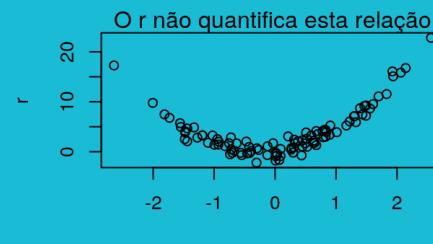
Relação linear negativa



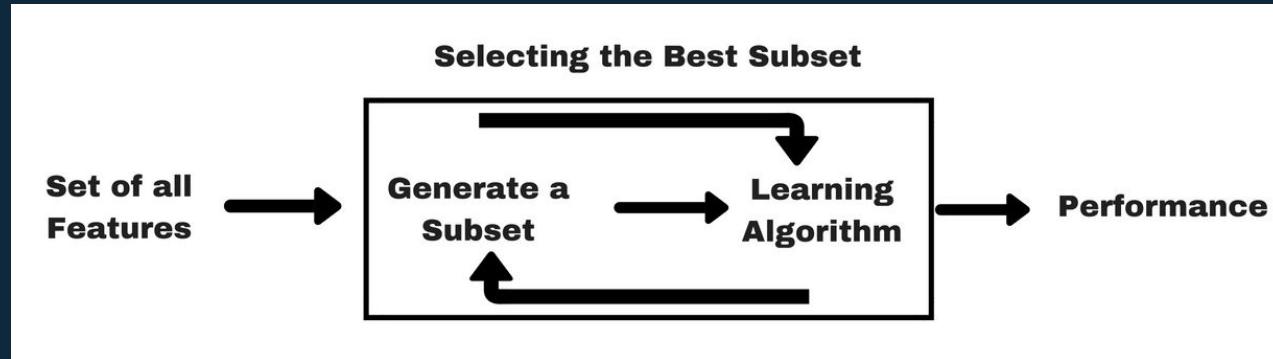
Ausência de relação



Relação não-linear



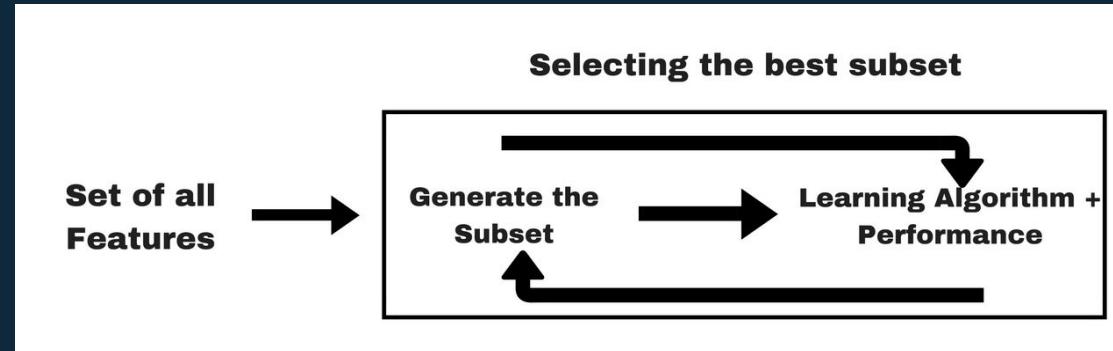
Wrapper method



- ◊ Forward Selection
- ◊ Backward Elimination
- ◊ Recursive Feature elimination

Qual a complexidade de todos os testes possíveis?

Embedded method

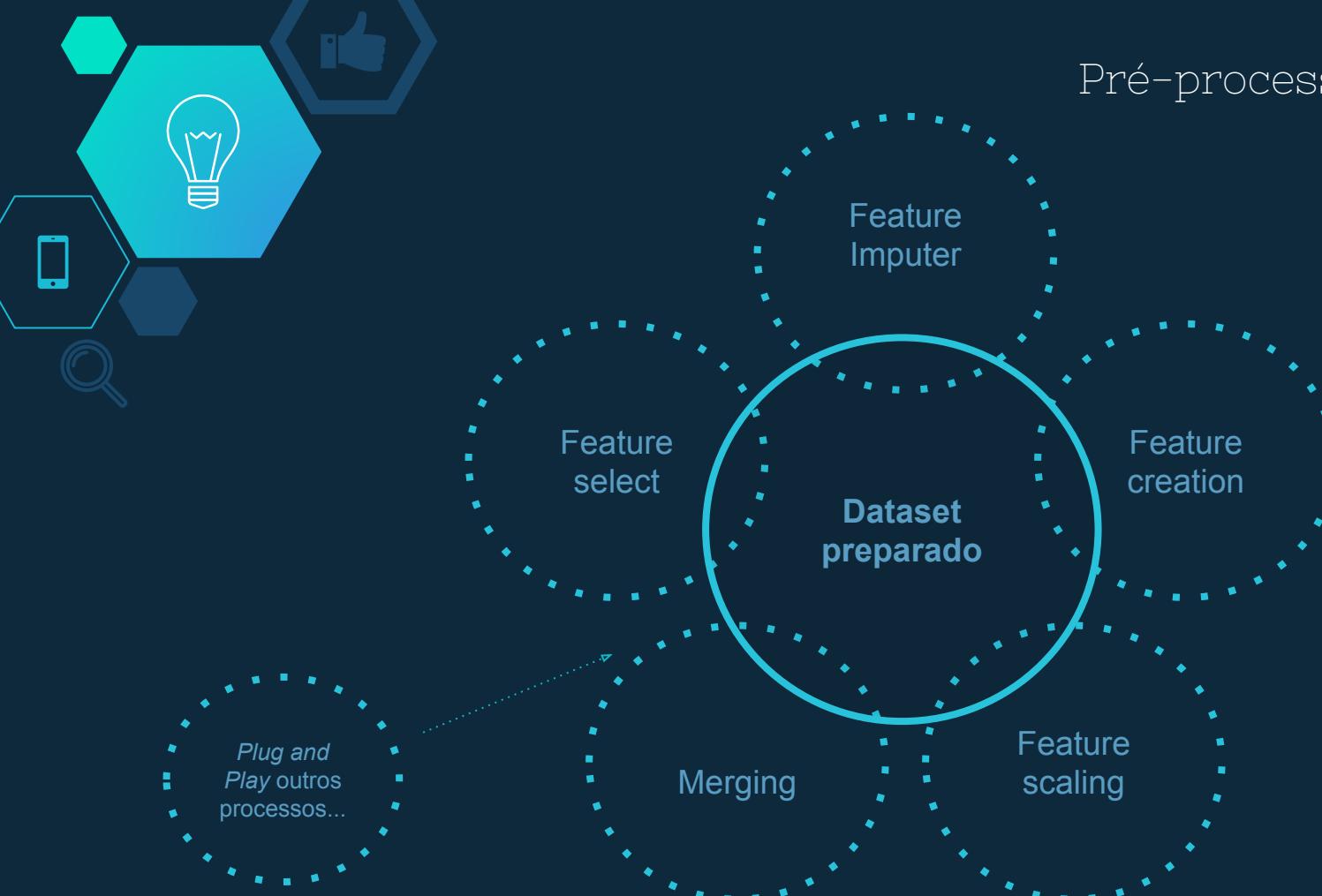


- ◊ Ganho da informação
 - Modelos baseado em árvore
- ◊ Lasso regression performs L1
- ◊ Ridge regression performs L2

Pré-processamento do dado



Pré-processamento do dado



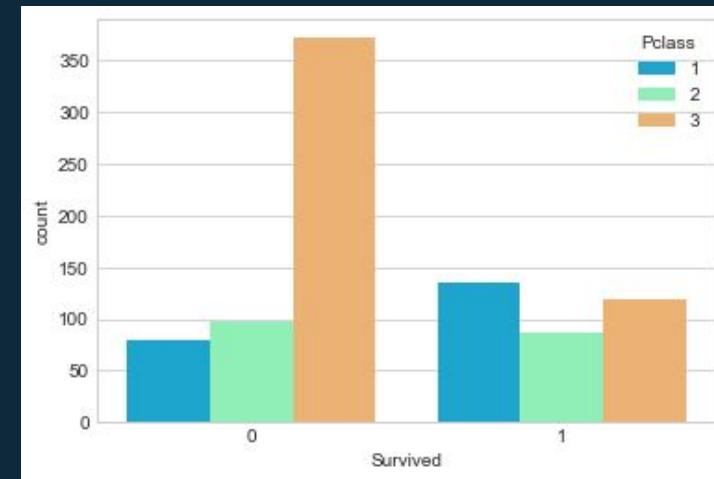
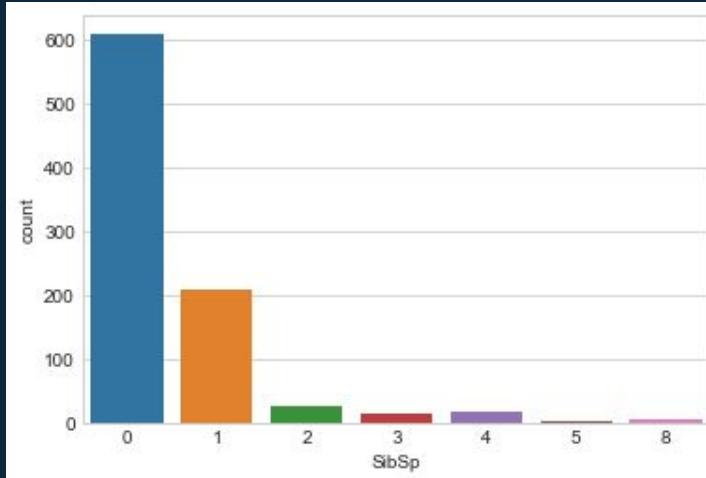


Qual o melhor
pré-processamento
do dado?



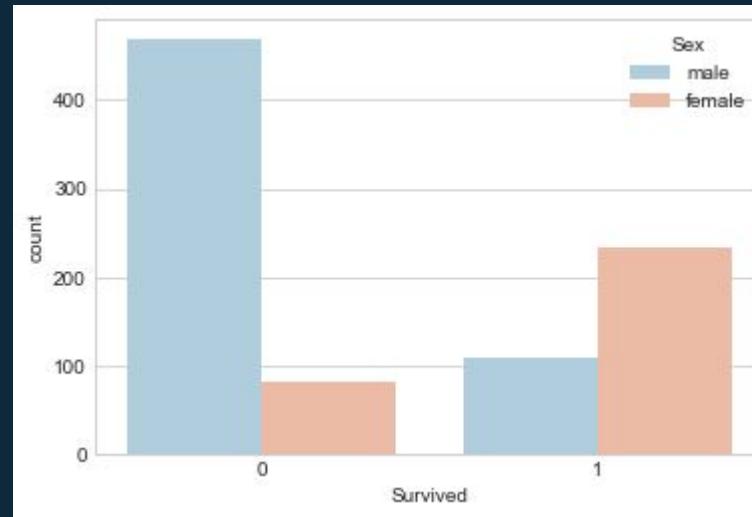
Hands on
<https://notepad.pw/kdduni7>

Insights (e.g.: ...)





Insights(e.g.: ...)





Aprendizagem Supervisionada





“São apresentadas ao computador exemplos de entradas e saídas desejadas, fornecidas por um “professor”. O objetivo é aprender uma regra geral que mapeia as entradas para as saídas.”

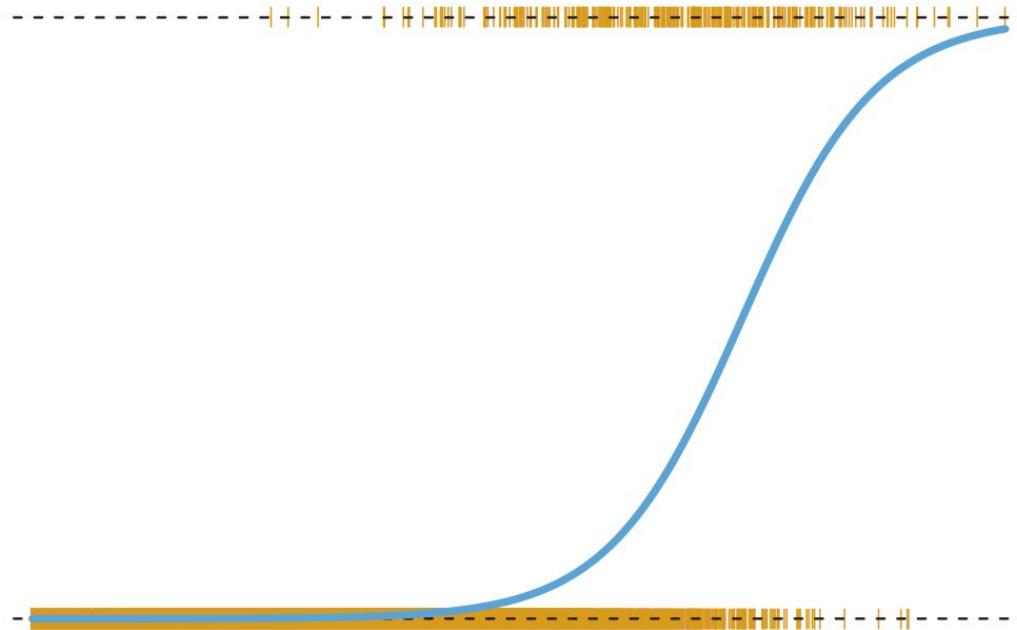
Conceitos

Algoritmo treinado sobre dados rotulados:

- ◊ Algoritmos de Classificação (e.g. tumor maligno e/ou benigno)
 - Prediz Valores discretos
 - Algoritmos: Árvore de Decisão, Regressão Logística, KNN, etc.
- ◊ Algoritmos Regressão linear (e.g. prever o preço de um imóvel)
 - Prediz Valores contínuos
 - Algoritmos: Regressão Linear e Polinomial, SVM Regressor, Árvore de Decisão

+ Exemplos

- ◊ Identificação de fraudes
- ◊ Detecção de epidemias
- ◊ Precisão de tratamentos
- ◊ Análise de sentimento
- ◊ Filtros de spam
- ◊ Cálculo de empréstimo



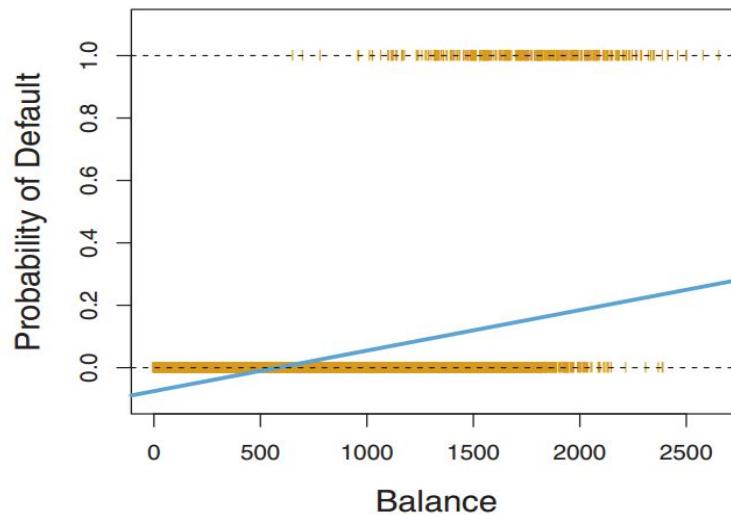
Regressão Logística

Definição e objetivo

- ◊ É uma forma especial de regressão formulada para prever, a partir de um conjunto de variáveis preditoras, uma única variável dependente com duas (binária) ou mais categorias
- ◊ Gerar uma **função matemática** cuja resposta permita **estabelecer a probabilidade** de uma observação **pertencer a um grupo previamente determinado**, em razão do comportamento de um conjunto de variáveis independentes

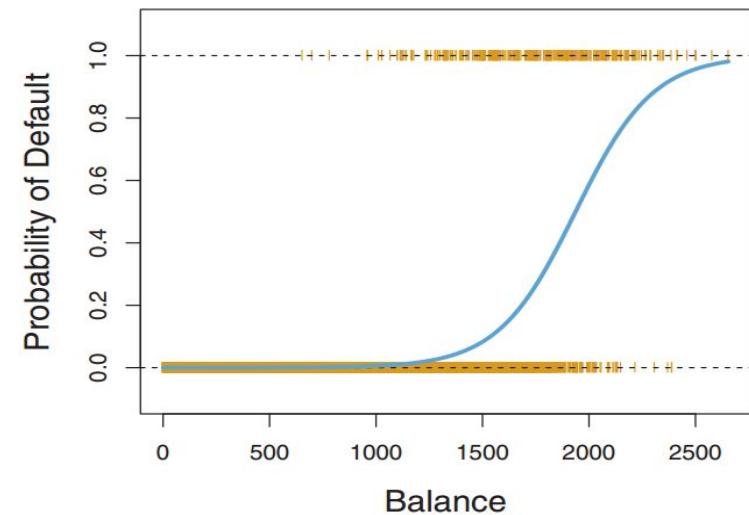


Regressão Linear



$$p(X) = \beta_0 + \beta_1 X.$$

Regressão Logística

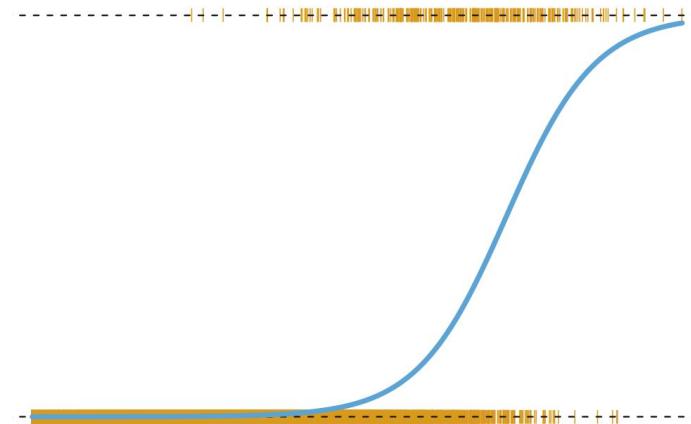


$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$



Função logística (sigmoid)

A função logística sempre produzirá uma curva em forma de S independente do valor de X, dando uma previsão sensata da probabilidade de uma amostra pertencer ou não à uma classe



$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$



Odds (Chances)

- ◊ A quantidade $p(X)/(1 - p(X))$ é chamada de odds e pode assumir qualquer valor entre 0 (zero) e ∞ (infinito).
 - Valores de odds próximos de 0 (zero) indicam uma predição muito baixa
 - Valores próximos de ∞ (infinito) probabilidades muito altas
- ◊ As odds são tradicionalmente usadas, no lugar de probabilidades, em corridas de cavalos, uma vez que eles se relacionam mais naturalmente com a estratégia de apostas

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$



Função Logit

- ◆ Há uma transformação da variável dependente, isto é, ela é convertida em uma razão de probabilidades em uma variável de base logarítmica (transformação logística).

$$\log \left(\frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X$$

- ◆ $p(X)$ é a probabilidade de ocorrer um evento;
- ◆ $(1 - p(X))$ é a probabilidade de não ocorrer o evento;
- ◆ $p/(1-p)$ a razão de probabilidades;
- ◆ X são as variáveis independentes;
- ◆ β são os coeficientes estimados;



Máxima Verossimilhança

- ◆ O objetivo é encontrar β_0 e β_1 de forma que estas estimativas para o modelo de $p(X)$, produza um número perto de 1 (um) para todos os indivíduos que satisfazem a variável dependente, e um número próximo de 0 (zero) para todos os indivíduos que não satisfazem. Essa intuição pode ser formalizada usando uma equação matemática chamada **função de verossimilhança**

i) Função de Verossimilhança $\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$

- ii) Probabilidade de $p(X)$:
Função logística

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Exemplo

Número de partos	Número de óbitos neonatais	Percentual de óbitos neonatais	<u>Nascimento < 37 semanas</u> Pré-termo(X1)	<u>Mães com diabetes</u> (X2)
10	1	10,00%	1	1
14	2	14,29%	0	1
181	16	8,84%	1	0
670	5	0,75%	0	0

Exemplo

Número de partos	Número de óbitos neonatais	P	1-P	Odds W=P/(1-P)	Logit(P) Ln(W)
10	1	10,00%	0,9	0,111	-2,200
14	2	14,30%	0,857	0,167	-1,790
181	16	8,80%	0,912	0,097	-2,330
670	5	0,70%	0,993	0,008	-4,890

Exemplo

- ◆ Odds: [0,111; 0,167; 0,097; 0,008]
- ◆ Valores Lógite: [-2,200; -1,790; -2,330; -4,890]
 - $\text{Logit}(P) = \beta_0 + \beta_1.X_1 + \beta_2.X_2$
 - Aplicando a máxima verossimilhança, temos que:
 - $\text{Logit}(P) = -4,15 + 1,076X_1 + 1,617X_2$



Exemplo

- ◊ $\text{Logit}(P_{\text{diab}}) = -4,15 + 1,076X_1 + 1,617$
- ◊ $\text{Logit}(P_{\sim \text{diab}}) = -4,15 + 1,076X_1$
- ◊ $\text{Logit}(P_{\text{diab}}) = \text{Logit}(P_{\text{não diab}}) + 1,617$
 - $\text{Logit}(P_{\text{diab}}) - \text{Logit}(P_{\text{não diab}}) = 1,617$
 - $\ln(P) - \ln(1-P) = 1,617$
 - $\ln[P / (1-P)] = 1,617$
 - $[P / (1-P)] = 5,04$

Odd de $P(O_{Ni=1} | M_{\text{diab}=1})$

$$\log \left(\frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X$$

Conclusões

- ◆ Os resultados podem ser interpretados em termos de probabilidade, possibilitando que seja medida a probabilidade de um indivíduo assumir determinada classificação (1|0) em função de um conjunto de atributos
- ◆ Os coeficientes indicam a importância de cada variável independente para a ocorrência do evento;

Observações

- ◊ Determinar a variável dependente:
 - Deve ser categórica
 - Se a variável for métrica ela deve ser convertida em um valor discreto
 - Mutuamente excludente pelas variáveis independentes
- ◊ Cada grupo de variável dependente deveria ter um único perfil nas variáveis independentes usadas
- ◊ Se dois grupos tem perfis semelhantes, quanto a uma variável independentes a RL não será capaz de estabelecer inequivocamente o perfil de cada grupo

Observações

- ◊ Tamanho geral da amostra:
 - [Ideal] 20 observações para variável independente
 - [Mínimo] 5 observações para variável independente

- ◊ Tamanho da amostra por categoria:
 - O menor número de amostras de uma categoria deve no mínimo exceder o número de variáveis independentes;
 - Os tamanhos relativos dos grupos não podem variar muito



Hands-On





Árvores de Decisão



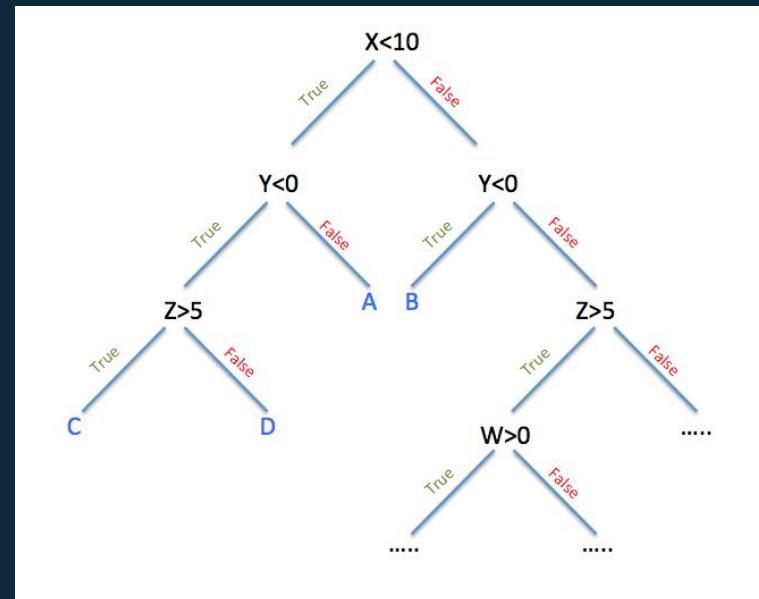


Árvore de Decisão



Information Gain

- ◊ $GI(x) = E(\text{Classe}) - E(x)$
 - X = atributo
 - $E(\text{Classe})$ é a entropia da classe no dataset
 - $E(x)$ é a entropia do atributo.
É a soma ponderada das entropias de suas partições



Information Gain

ESCOLA	IDADE	LABEL(Bolsa?)
Part_bolsa	>18	Não
Particular	<=18	Não
Part_bolsa	<=18	Sim
Particular	>18	Não
Pública	<=18	Sim
Pública	>18	Sim
Part_bolsa	>18	Não
Part_bolsa	<=18	Sim



Entropia da classe

◊ Entropia da classe:

- $E(\text{Bolsa}) = -\sum P_i \log_2 P_i$
 - ◊ 4 bolsistas
 - ◊ 4 não bolsistas
 - ◊ 8 amostras
- $E(\text{Bolsa}) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$
- $E(\text{Bolsa}) = -0,5 \log_2 0,5 - 0,5 \log_2 0,5$
- $E(\text{Bolsa}) = 1$

Entropia do atributo

<u>ESCOLA</u>	IDADE	LABEL(Bolsa?)
Part_bolsa	>18	Não
Particular	<=18	Não
Part_bolsa	<=18	Sim
Particular	>18	Não
Pública	<=18	Sim
Pública	>18	Sim
Part_bolsa	>18	Não
Part_bolsa	<=18	Sim

Bolsa?	PB	PA	PU
Não	2	2	0
Sim	2	0	2

Information Gain

- ◊ Entropia do atributo (Escola):
 - $E(\text{Escola}) = -\sum P_i \log_2 P_i$
 - $P_1(\text{bolsa=sim}|\text{Esc=PU}) = 2/2 = 1$
 - $P_2(\text{bolsa=nao}|\text{Esc=PU}) = 0/2 = 0$
 - $E_{\text{esc}}(\text{Esc=PU}) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$
 - $E_{\text{esc}}(\text{Esc=PU}) = -1 \log_2 1 - 0 \log_2 0$
 - **$E_{\text{esc}}(\text{Esc=PU})=0$**

Bolsa?	PU	PB	PA
Não	0	2	2
Sim	2	2	0

Information Gain

◆ Entropia do atributo (Escola):

- $E(\text{Escola}) = -\sum P_i \log_2 P_i$
 - $P_1(\text{bolsa=sim}|\text{Esc=PB}) = 2/4 = 1/2$
 - $P_2(\text{bolsa=nao}|\text{Esc=PB}) = 2/4 = 1/2$
- $E_{\text{esc}}(\text{Esc=PB}) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$
- $E_{\text{esc}}(\text{Esc=PB}) = -0,5 \log_2 0,5$
 $-0,5 \log_2 0,5$
- **$E_{\text{esc}}(\text{Esc=PB})=1$**

Bolsa?	PU	PB	PA
Não	0	2	2
Sim	2	2	0

Information Gain

◇ Entropia do atributo (Escola):

- $E(\text{Escola}) = -\sum P_i \log_2 P_i$
 - $P_1(\text{bolsa=sim}|\text{Esc=PA}) = 0/2 = 0$
 - $P_2(\text{bolsa=nao}|\text{Esc=PA}) = 2/2 = 1$
- $E_{\text{esc}}(\text{Esc=PA}) = -P_1 \log_2 P_1 - P_2 \log_2 P_2$
- $E_{\text{esc}}(\text{Esc=PA}) = -0 \log_2 0 - 1 \log_2 1$
- **$E_{\text{esc}}(\text{Esc=PA}) = 0$**

Bolsa?	PU	PB	PA
Não	0	2	2
Sim	2	2	0

Information Gain

- ◇ Entropia do atributo (Escola):
 - $E(Esc=PU) = 0$
 - $E(Esc=PB) = 1$
 - $E(Esc=PA) = 0$
 - ◇ $E(Esc) = P(Esc=PU)*E(Esc=PU) + P(Esc=PB)*E(Esc=PB) + P(Esc=PA)*E(Esc=PA)$
 - ◇ $E(Esc) = (2/4)*0 + (4/8)*1 + (2/8)*0$
 - ◇ $E(Esc) = 0,5$
- Fazendo todos esses cálculos para a idade temos:
- ◇ $E(Idade) = 0,81$

Information Gain

- ◊ $E(Esc) = 0,5$
- ◊ $E(Idade) = 0,81$
- ◊ $GI(Esc) = E(Bolsa) - E(Esc) = 1 - 0,5 = 0,5$
- ◊ $GI(Idade) = E(Bolsa) - E(Idade) = 1 - 0,811 = 0,189$

Qual o melhor atributo?

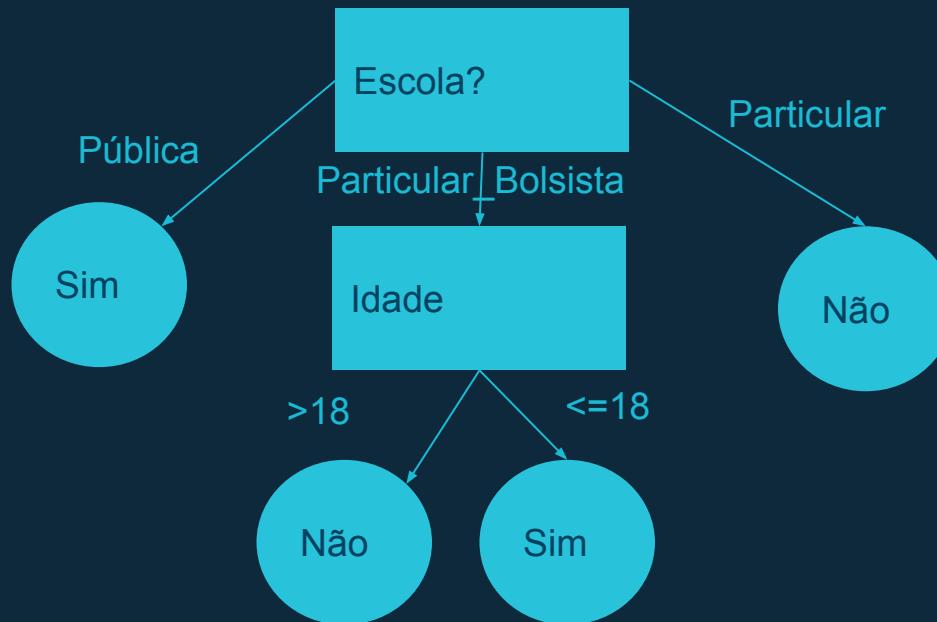
Algoritmo de Indução

Algoritmo de indução:

- 1) Escolher uma feature
- 2) Estender a árvore adicionando um ramo para cada valor do atributo
- 3) Filtrar as amostras de acordo com o valor do atributo e enviar as amostras para a folha
- 4) Para cada folha:
 - a) Se as amostras forem da mesma classe, associar a folha.
 - i) Senão, repetir os passos de 1 até 4

Qual o melhor atributo?

Indução da árvore



Considerações

- ◊ GI tem um bias que favorece a escolha de atributos com muitos valores;
- ◊ Para minimizar o *overfitting* deve-se:
 - Aplicar procedimentos de poda
 - Definir bem os hiperparâmetros
 - Selecionar atributos a priori, etc.



Random Forest

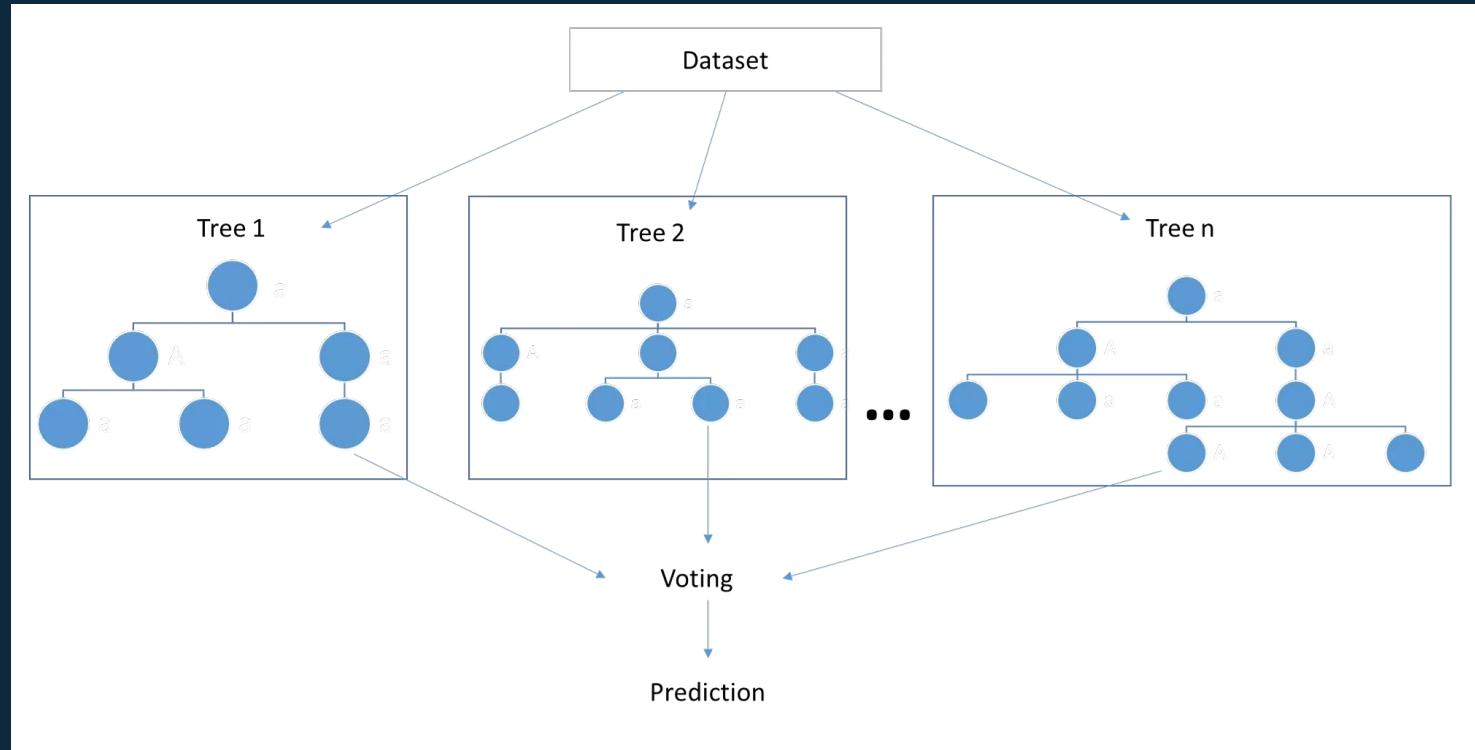
- ◆ A “floresta” que ele cria é uma combinação (ensemble) de árvores de decisão
- ◆ Treinados com o método de bagging, (amostras diferentes da base de dados que são usadas para aprender hipóteses diferentes)
- ◆ Busca a melhor característica em um subconjunto aleatório das características



Random Forest

- ◆ A previsão final para um exemplo de teste é a média da previsão de cada hipótese
- ◆ Cria diversidade, o que geralmente leva a geração de modelos melhores.
- ◆ Muito bom para se medir a importância relativa de cada característica (feature) para a predição

Random Forest



Considerações

- ◊ Vantagens:
 - Poder ser utilizado tanto para regressão quanto para classificação
 - É fácil visualizar a importância relativa que ele atribui para cada característica na suas entradas
 - O número de hiperparâmetros não é tão grande e são fáceis de serem compreendidos.
 - Diminui o overfitting se comparado a árvore de decisão

Considerações

◆ Desvantagens:

- Uma quantidade grande de árvores pode tornar o algoritmo lento e ineficiente para predições em tempo real.
- Muito lentos para fazer predições depois de treinados (São rápidos para treinar)
- Uma predição com mais acurácia requer mais árvores, o que faz o modelo ficar mais lento

Hands-on - <https://notepad.pw/kdduni7>

Coding

vector illustration

</>

dreamstime.





Métricas de avaliação



Área Sob Curva ROC

- ◊ Criada por engenheiros elétricos e de sistemas de radar durante a Segunda Guerra Mundial para detectar objetos inimigos em campos de batalha
- ◊ Os algoritmos de classificação produzem um valor situado dentro de um determinado intervalo contínuo, como $[0;1]$, é necessário definir um ponto de corte, ou um limiar de decisão, para se classificar e contabilizar o número de

Área Sob Curva ROC

- ◊ Este limiar pode ser selecionado arbitrariamente, a melhor prática para se comparar o desempenho de diversos sistemas é estudar o efeito de seleção de diversos limiares sobre o resultado das previsões.

Área Sob Curva ROC

- ◊ Para cada ponto de corte são calculados valores de sensibilidade e especificidade:
- ◊ **Sensibilidade:**
 - A proporção de verdadeiros positivos: a capacidade do sistema em predizer corretamente a condição para casos que realmente a têm.
 - $\text{SENS} = \text{ACERTOS POSITIVOS} / \text{TOTAL DE POSITIVOS}$
 $= \text{VP} / (\text{VP} + \text{FN})$

Área Sob Curva ROC

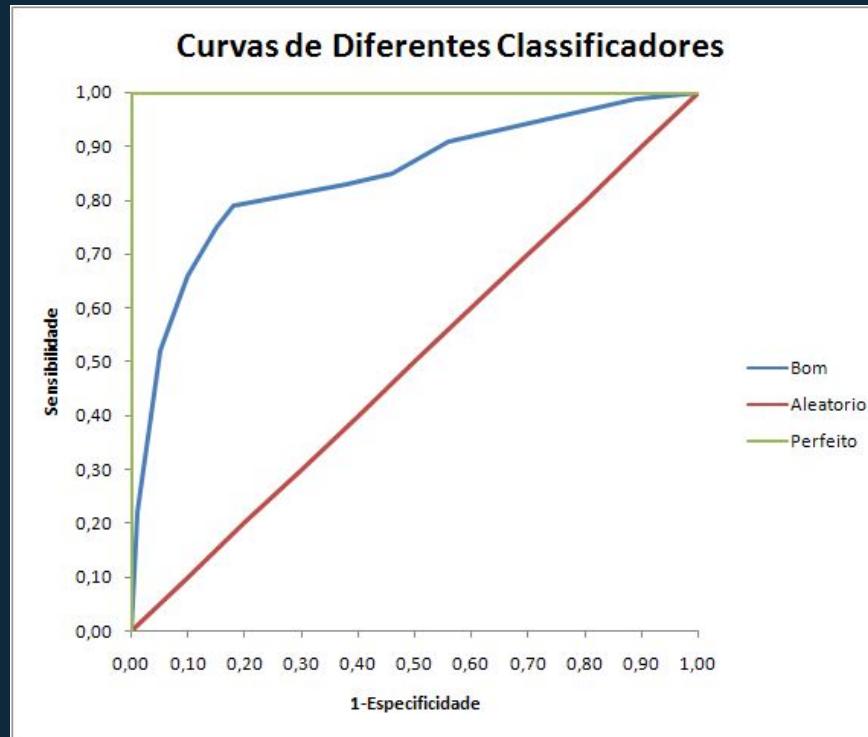
◆ Especificidade:

- A proporção de verdadeiros negativos: a capacidade do sistema em predizer corretamente a ausência da condição para casos que realmente não a têm.
- $SPEC = ACERTOS\ NEGATIVOS / TOTAL\ DE\ NEGATIVOS$

$$= VN / (VN + FP)$$

	Pred. Pos.	Pred. Neg.
Label Pos.	VP	FP
Label Neg.	FN	VN

Área Sob Curva ROC





Análise do Bias e Variance

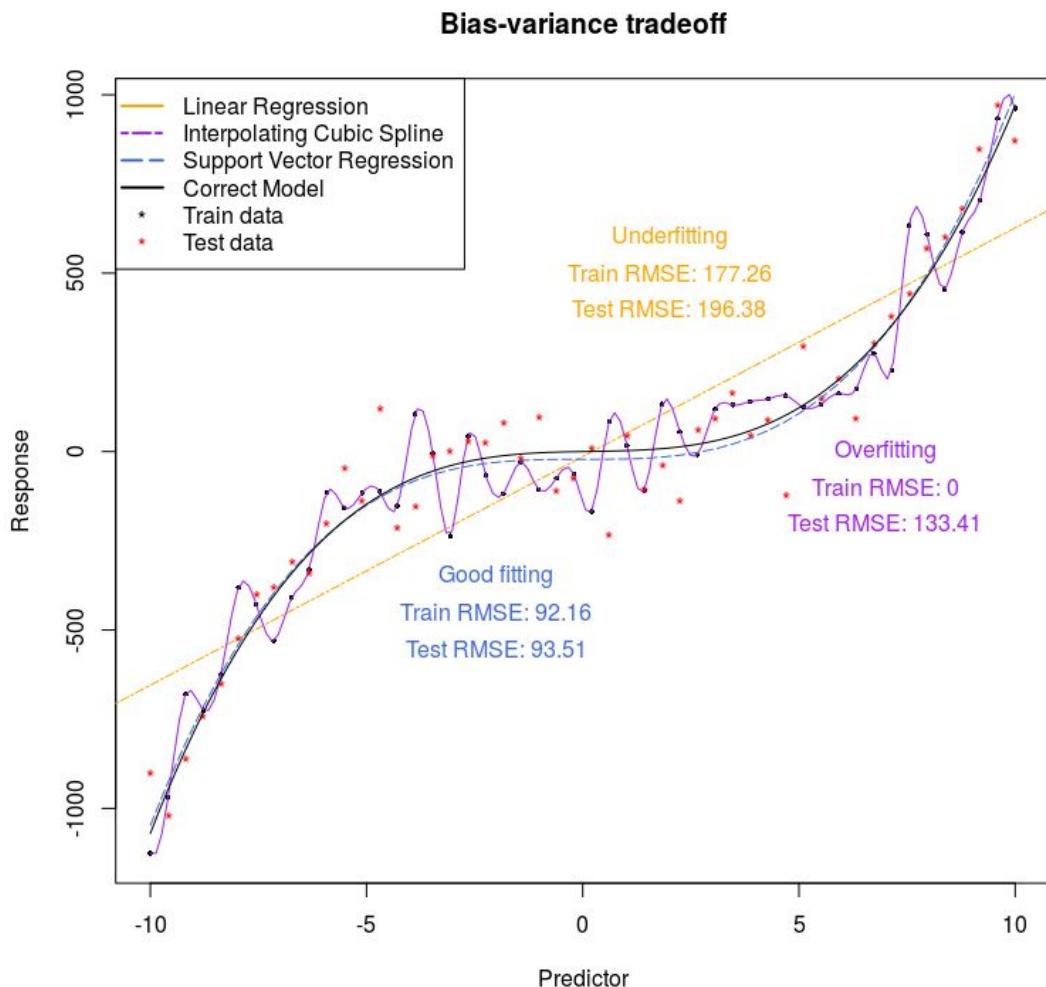


Bias vs Variance

- ◊ **Underfitting** - Utilizar um modelo simples que é bem generalizável, **mas não reduz consideravelmente** o erro de previsão no train set. Nesse caso estamos optando por um modelo com viés mais alto, mas variância baixa.
- ◊ **Overfitting** - Utilizar um modelo complexo que é capaz de **reduzir consideravelmente** o erro de previsão no train set, mas ao mesmo tempo **não é tão generalizável a ponto de apresentar um bom resultado no test set**. Nesse caso estamos optando por estimar um modelo com viés baixo e variância alta.



Bias vs Variance



Bias vs Variance

$$E[(y - \hat{f}(x))^2] = Bias[\hat{f}(x)]^2 + Var[\hat{f}(x)] + \sigma^2$$

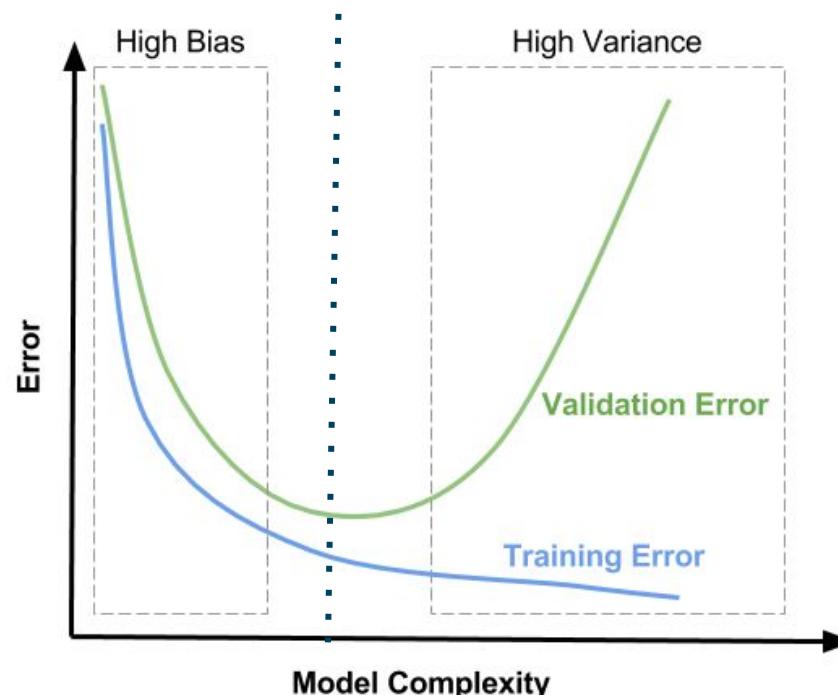
$$Bias[\hat{f}(x)] = E[\hat{f}(x) - f(x)],$$

$$Var[\hat{f}(x)] = E[\hat{f}(x)^2] - E[\hat{f}(x)]^2.$$

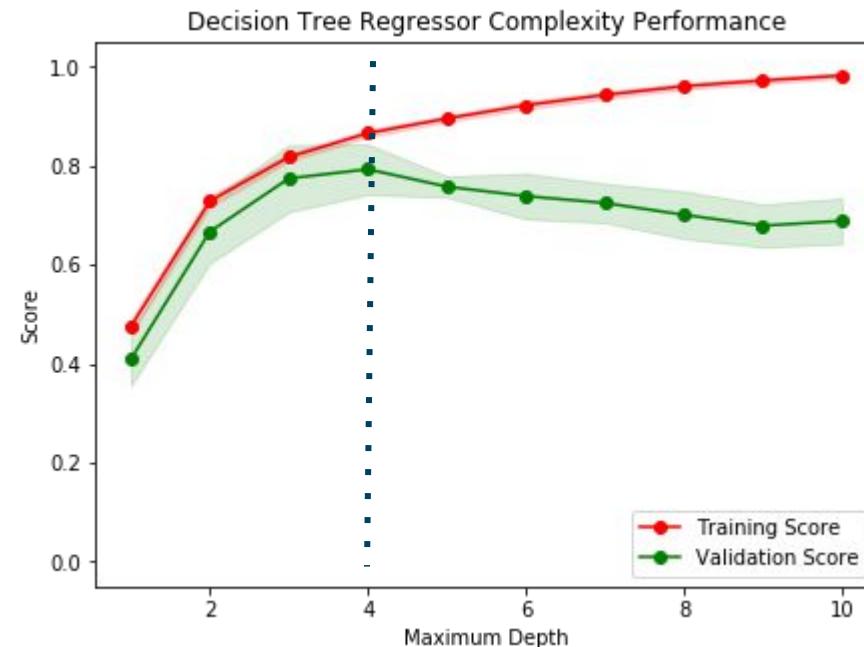
Bias vs Variance

- ◊ O objetivo é escolher a $f(x)$, um modelo, próxima do ideal, visto que tanto o viés quanto a variância aumentam o erro de previsão.
- ◊ Obviamente a escolha entre bias e variance é um tradeoff, e o ideal é permanecermos em um meio termo entre um modelo complexo e um bem generalizável.

Bias vs Variance



Bias vs Variance





Hands-On



Modelo encadeado

O cliente atende aos nossos critérios?

Dados do
Aluno
[.....]

Modelo de
atendimento aos
critérios do banco

Ele vai pagar o empréstimo?

Modelo de classificação de
se vai pagar ou não

Empresa
ou não

Modelo de empréstimo do nosso banco UNI7_BANK

KNN (para regressão)

k-nearest neighbors





Introdução

- ◊ O meu comportamento é influenciado (ou caracterizado) por todos esses fatores, logo olhar para os vizinhos mais próximos de mim nessa dimensão parece ser melhor que olhar para toda a população
- ◊ Essa é a idéia por trás do k-Vizinhos mais próximos (kNN)



O Algoritmo KNN

- ◊ O kNN é um dos modelos de predição mais simples que existem. Ele utiliza apenas:
 - Alguma noção de distância (similaridade)
 - A suposição que pontos que estão próximos uns dos outros são semelhantes
- ◊ Para a predição de um novo ponto, observa-se apenas os pontos mais próximos

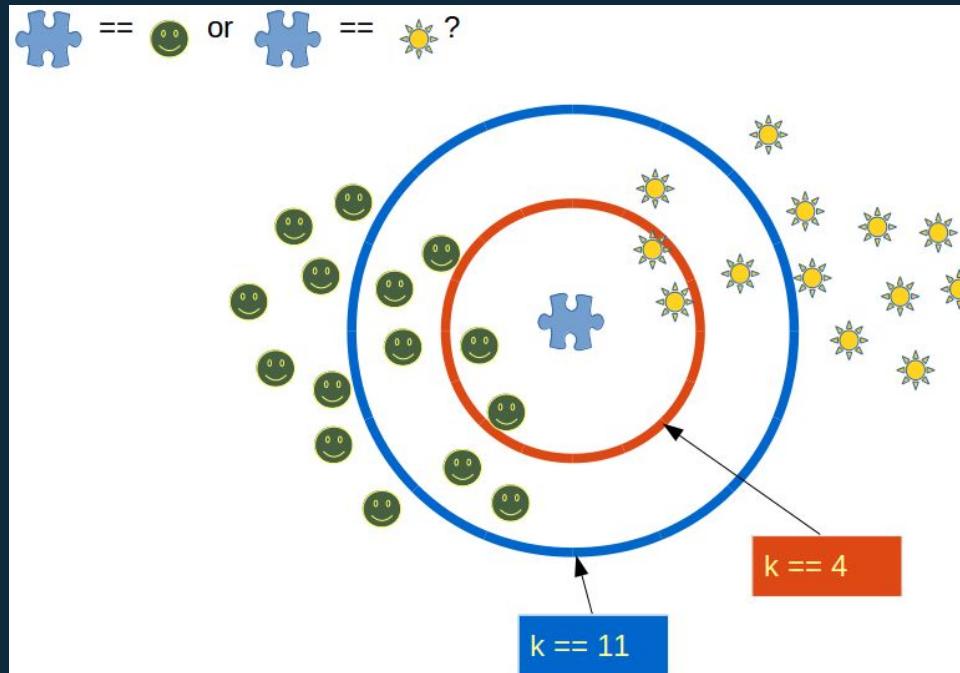
O Algoritmo KNN

- ◊ Não ajuda a entender o fenômeno observado: Não há como recuperar os pesos das features, por exemplo
- ◊ Armazena-se uma base de exemplos (instances) que é usada para realizar a classificação de uma nova query (exemplo não visto);

O Algoritmo KNN

- ◊ Em geral temos os dados rotulados
- ◊ Para um ponto novo, procura-se os k pontos mais próximos a ele e olha-se os seus labels para decidir qual será o label do novo
- ◊ E se der empate? Temos várias opções:
 - Escolher um dos ganhadores aleatoriamente
 - Atribuir pesos aos votos baseado na distância
 - Reduzir k até encontrar um vencedor

KNN - Exemplo





Considerações

- ◊ Podemos favorecer as amostras mais similares
- ◊ Podemos aplicar uma média ponderada para considerar os vizinhos mais próximos
- ◊ A predição é sobre demanda
- ◊ kNN tem problemas com dimensões maiores graças a maldição da dimensionalidade, que se resume a idéia de que espaços de dimensão alta são vastos.



Considerações

- ◊ Pontos nesses espaços tendem a não estar próximos
- ◊ Cada dimensão é uma chance de aumentar a distância
- ◊ Normalmente usa-se a redução de dimensionalidade antes de usar o kNN

R^2 – Métrica de avaliação

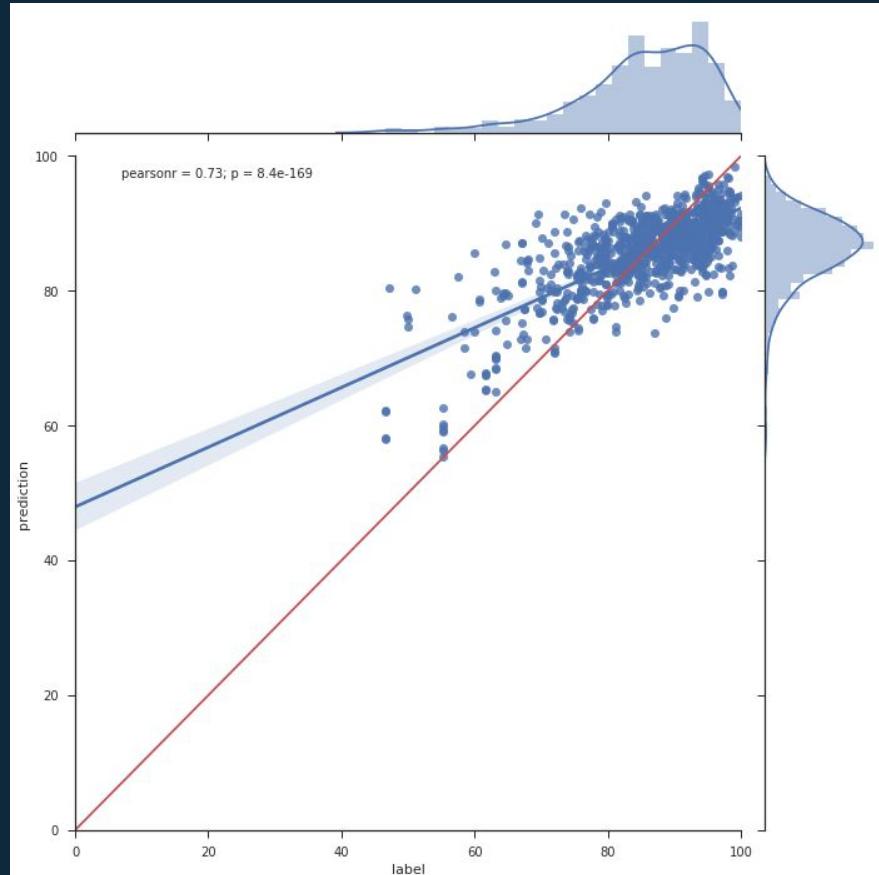


O coeficiente de determinação é uma medida de ajustamento de um modelo estatístico linear generalizado, como a regressão linear, em relação aos valores observados.

O R^2 varia entre 0 e 1, indicando, em percentagem, o quanto o modelo consegue explicar os valores observados. Quanto maior o R^2 , mais explicativo é o modelo, melhor ele se ajusta à amostra.

Por exemplo, se o R^2 de um modelo é 0,8234, isto significa que 82,34% da variável dependente consegue ser explicada pelos regressores presentes no modelo.

Visualização Predicted vs Observed



Hands-on

Coding

vector illustration

</>

dreamstime.





+ Handson - desafio

- Crie um pipeline para o tratamento dos dados (<http://queirozf.com/entries/scikit-learn-pipeline-examples>)
- Realize predição sobre os dados de validação (titanic_test.csv)
- Vamos retornar ao trabalho do Titanic e fazer nosso modelo retornar no mínimo 0,94 para área sob a curva ROC para os dados de validação.



Aprendizagem não supervisionada





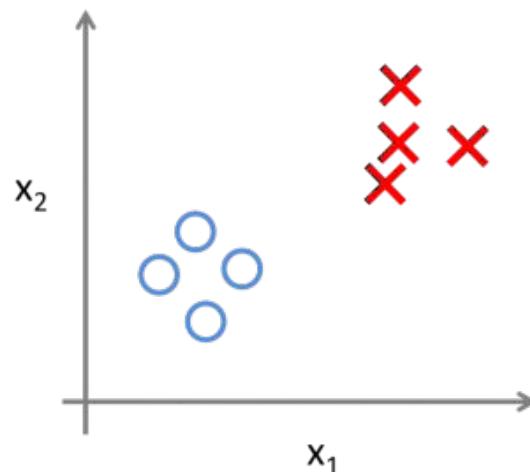
“Você pode ter os dados sem informações, **mas** você não pode ter informações sem dados”.



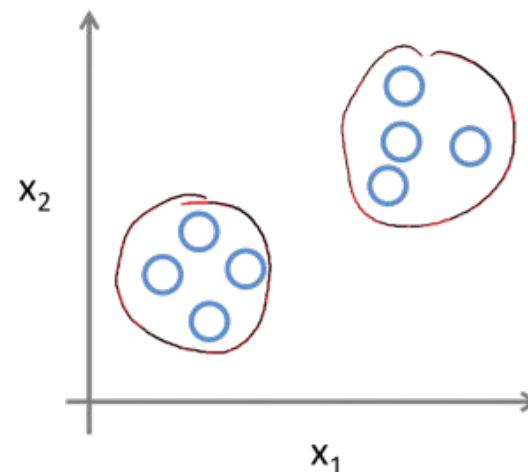
Aprendizagem não supervisionada

Introdução

Supervised Learning



Unsupervised Learning



Introdução

- ◊ Classificar novos dados não rotulados
- ◊ Formação de grupos
- ◊ Exemplos:
 - Agrupamento de clientes
 - Cocktail party



Aprendizagem não supervisionada

Introdução

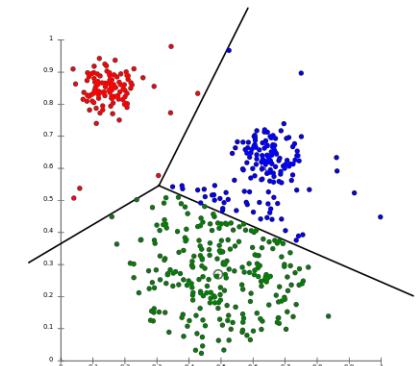
- ◊ Podemos ganhar alguma percepção da natureza (ou estrutura) dos dados.
- ◊ Podemos Identificar padrões implícitos
- ◊ Procurar por outliers e detectar anomalias

Introdução

- ◊ Objetivo: Amostras dentro de um mesmo cluster sejam muito parecidos, e amostras em clusters diferentes sejam distintos entre si.
- ◊ São utilizados medidas de similaridade entre as amostras
- ◊ Exemplos de análise de cluster são:
 - Hiéraquico
 - Baseado no centroíde
 - Baseado na densidade

KMEANS

Um dos mais conhecidos algoritmo de clusterização



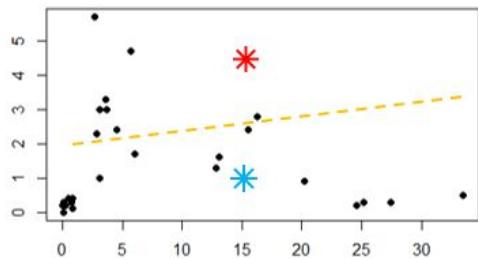


Introdução

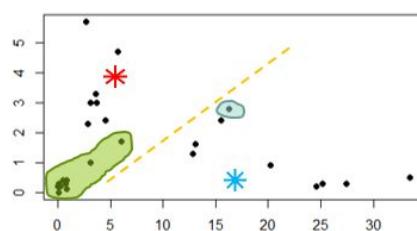
- ◊ Baseado em centroíde (representante)
- ◊ Os centróides (pontos centrais do grupo)
- ◊ **K-means** tem como objetivo agrupar os dados em $C1, C2, \dots, Ck$ clusters, de acordo com a medida de distância média (**means**) dos pontos, o qual define os centróides
- ◊ As partições têm a finalidade de minimizar a distância de todos os pontos, de um cluster, aos seus respectivos centróides

Algoritmo

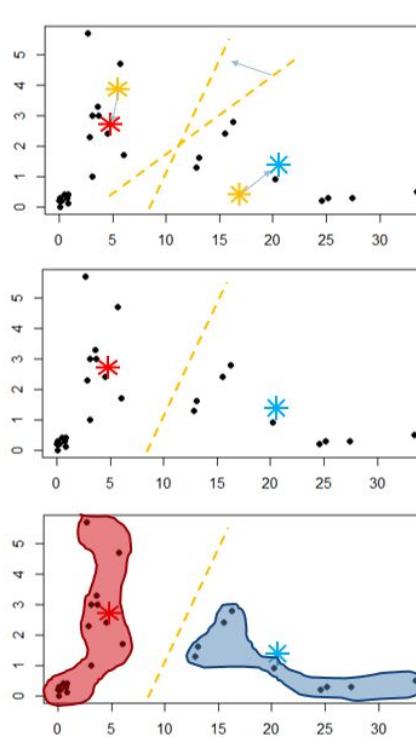
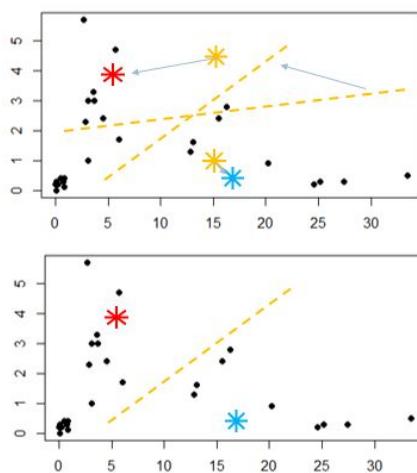
- ◊ Escolhe randomicamente K pontos para representar os centróides iniciais
- ◊ Agrupa todos os pontos aos seus centróides mais próximos, de acordo com a medida de distância.
- ◊ Verifica as seguintes situações:
 - Se houve mudanças de grupos/novas atribuições de pontos, calcula-se novamente o vetor central de um grupo (definindo um novo centróide) e volto para a etapa 2.
 - Caso não haja mudanças, o algoritmo é finalizado.



1. Defino $k=2$ centróides iniciais e defino os clusters

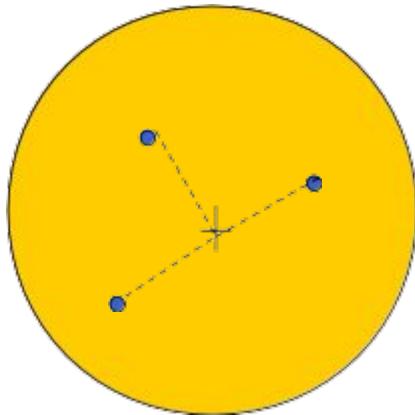


2. Recalcula o vetor central, definindo novos centróides, e assim atribuo os pontos aos clusters adequados.

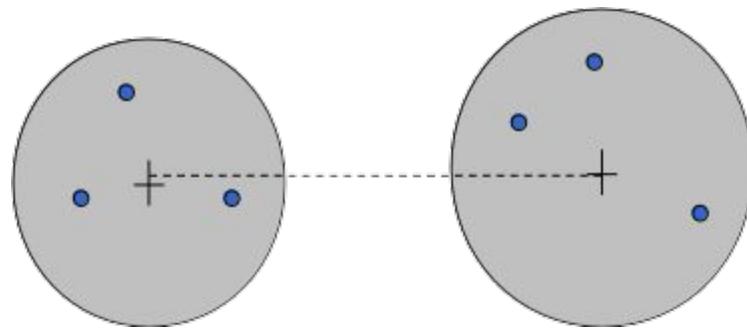


3. Nova disposição de centróides ocorre. Porém, nos últimos passos, não há mais movimentação de pontos, pois os centróides realmente representam os pontos centrais do grupo agora. Assim, temos o grupo “vermelho” e “azul”.

Como avaliar um cluster?



Coesão: mede a proximidade das amostras em relação ao centróide. É uma avaliação de um cluster (intra-cluster).



Separação: mede a qualidade de separação entre os clusters (inter-cluster).



Como escolher o K?

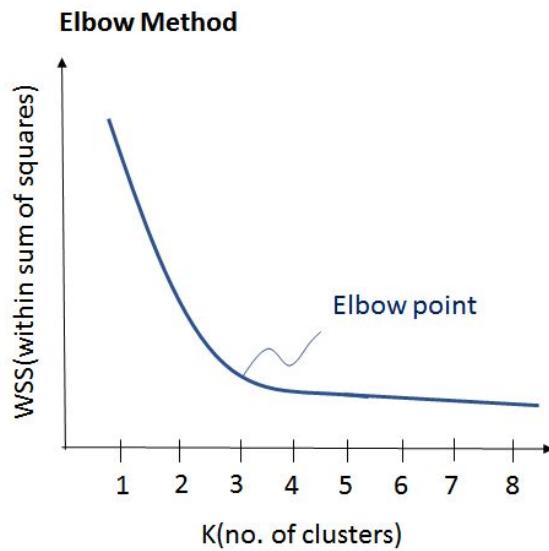
- ◆ Executar o algoritmo k-means para um intervalo de valores de k, calculando a Soma dos quadrados dos erros para cada clustering, obtida como:

$$D_r = \sum_{i=1}^{n_r-1} \sum_{j=i}^{n_r} \|d_i - d_j\|_2$$

Com x um ponto de dados (vetor) e **means(C_K)** o centróide do cluster K. Observe que x deve pertencer ao cluster K.



Como escolher o K?



- ◆ Executar o algoritmo k-means para um intervalo de valores de k (e.g.: [1:8])
- ◆ O objetivo é encontrar a variação intra-cluster e minimizá-la.

Aplicar o método de Elbow



Considerações

- ◊ Inicialização dos centroídes, pode provocar uma clusterização ruim
- ◊ Não há tratamento para os outliers, para isto há o **K-medoids**
- ◊ Aplique PCA se a dimensão das features for muito grande



Métricas de avaliação



Método da silhueta

- ◆ A silhueta mostra o quanto bem as amostras se posicionam dentro do cluster e quais meramente ficam em uma posição intermediária. Assim, cada cluster é representado por uma silhueta.
- ◆ O cálculo da **Largura Média de Silhueta (SWC)**, média da silhueta das amostras, é utilizado para selecionar o "melhor" número de clusters



Método da silhueta

- ◊ O Coeficiente de Silhueta é uma avaliação, em que uma pontuação mais alta de Coeficiente de Silhueta se relaciona a um modelo com clusters melhor definidos. O Coeficiente de Silhueta é definido para cada amostra e é composto por duas pontuações:
 - **a:** A distância média entre uma amostra e todos os outros pontos da mesma classe.
 - **b:** A distância média entre uma amostra e todos os outros pontos no próximo cluster mais próximo.
- ◊ O Coeficiente de Silhueta s para uma única amostra é então dado como:

$$s = \frac{b - a}{\max(a, b)}$$



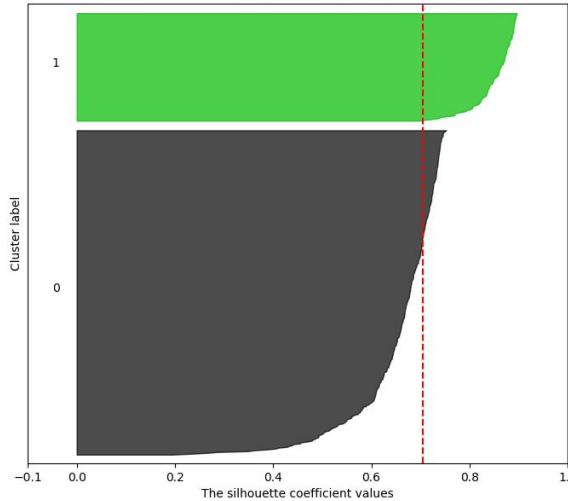
Método da silhueta

- ◊ O Coeficiente de Silhueta variam de [-1,1]:
 - **-1**: A amostra está mais próxima das amostras do cluster vizinho, mostrando que foi associada ao cluster atual erroneamente.
 - **0**: indica que a amostra está muito próxima do limite de decisão entre dois clusters vizinhos
 - **+1**: indica que a amostra está longe dos clusters vizinhos (está coeso)

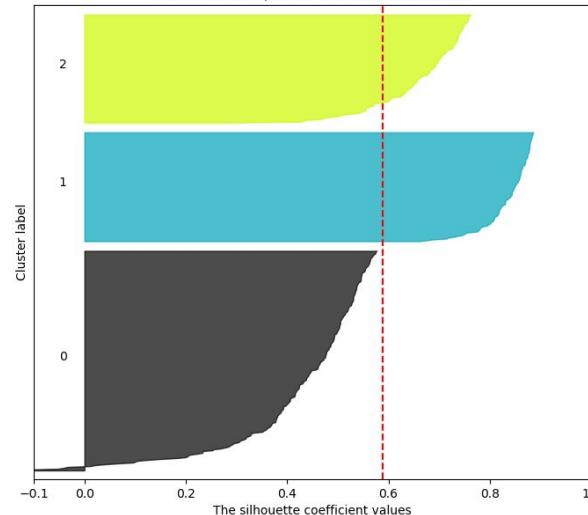


Método da silhueta

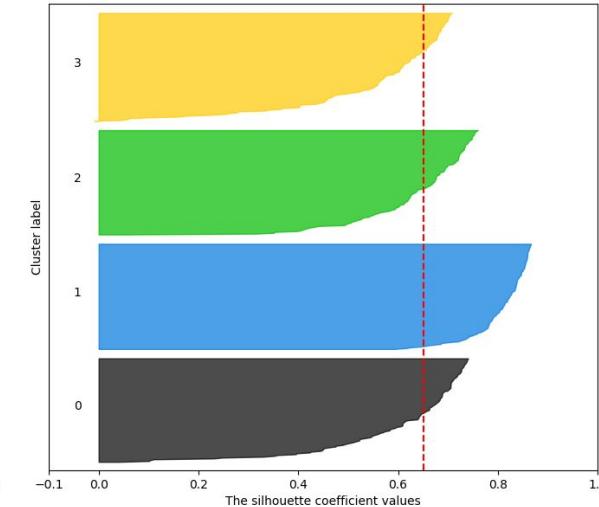
The silhouette plot for the various clusters.



The silhouette plot for the various clusters.



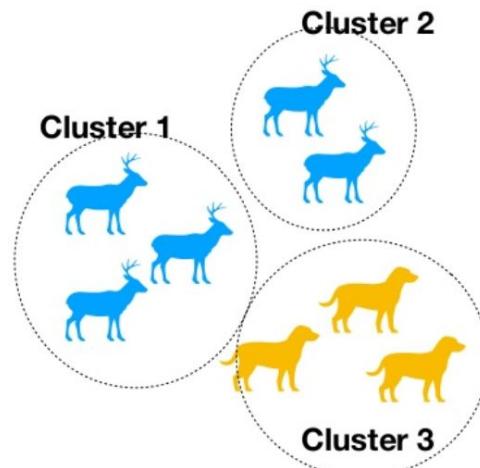
The silhouette plot for the various clusters.



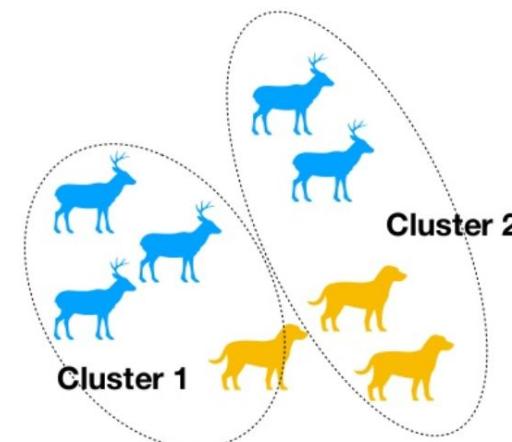


Homogeneidade

Cada cluster contém somente membros da sua classe



Good

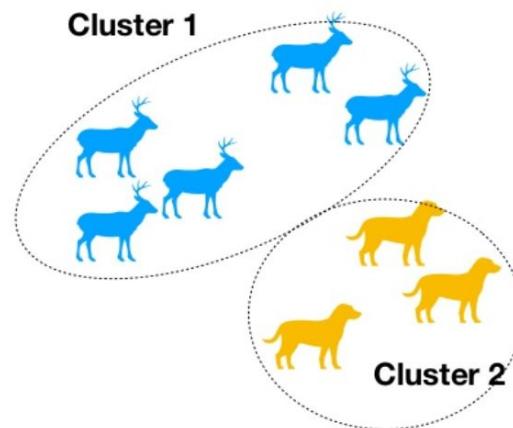


Bad

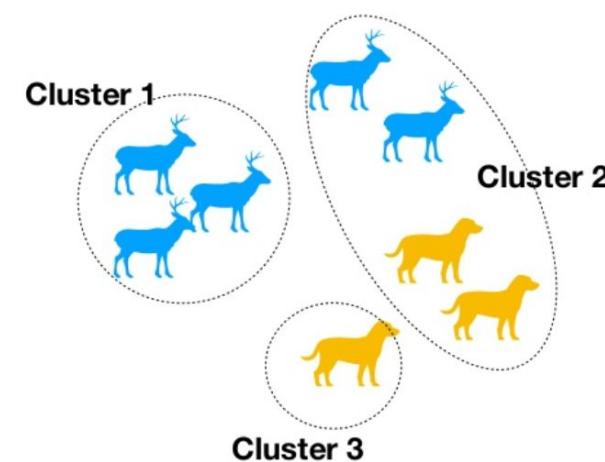


Completude

Todos os membros de uma determinada classe são atribuídos ao mesmo cluster.



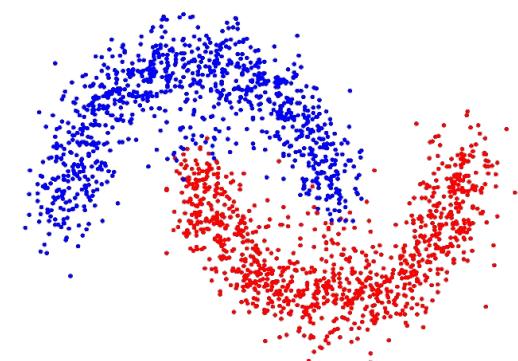
Good



Bad

DBSCAN

Density-based spatial clustering of applications with noise





Introdução

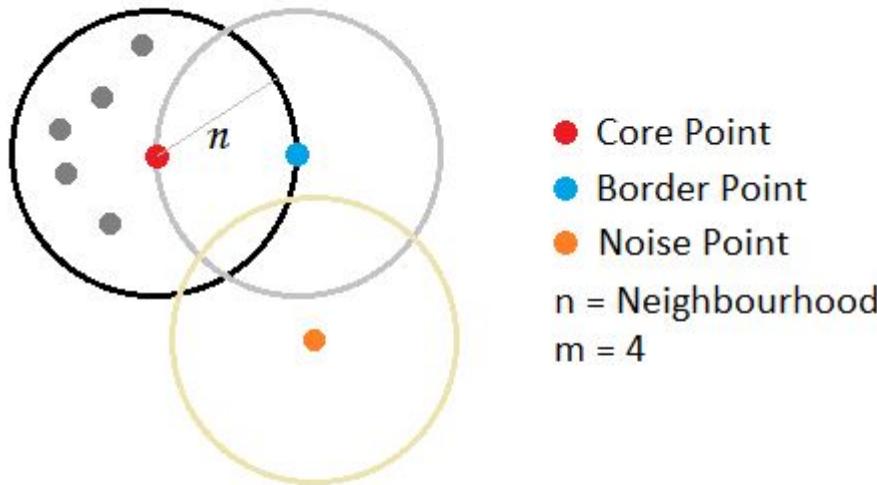
- ◊ Foi proposto em 1996
- ◊ Baseado em densidade
- ◊ Efetivo para **identificar** clusters de formato arbitrário e de diferentes tamanhos, **identificar e separar** os ruídos dos dados e **detectar clusters** “naturais” sem qualquer informação preliminar sobre os grupos.
- ◊ Há muitas variações deste algoritmo



Conceitos

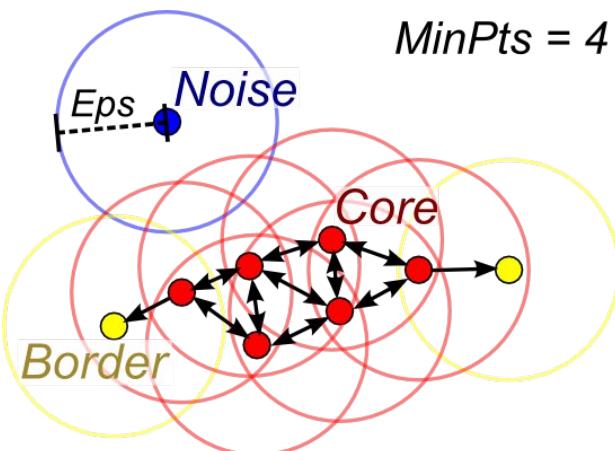
- ◊ **Vizinhança de um ponto (ϵ -vizinhança):**
 - $N(p) = \{ q \text{ em } D | \text{dist}(p,q) < \text{Raio} \}$
- ◊ **Core:**
 - Se a ϵ -vizinhança de um objeto p contém ao menos um número mínimo, MinPts, de objetos, então o objeto p é chamado de ponto central .
- ◊ **Border:**
 - Se a ϵ -vizinhança de um objeto p contém menos que MinPts, mas contém algum ponto central, então o objeto p é chamado de ponto de borda.
- ◊ **Noise:**
 - O ruído são o conjunto de pontos na base de dados D que não pertença a qualquer grupo Ci. Um objeto que não é ponto central nem ponto de borda, é ruído.

Conceitos



DBSCAN CLUSTERING

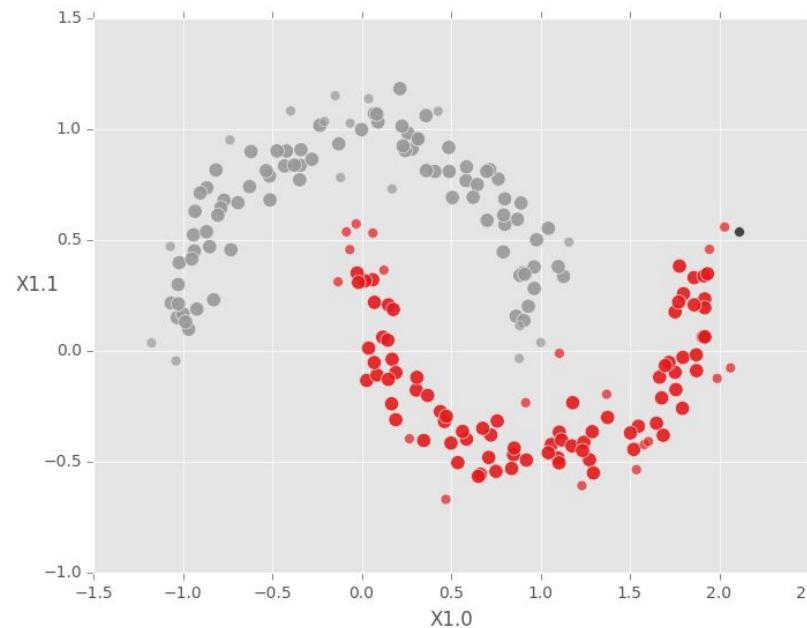
Algoritmo



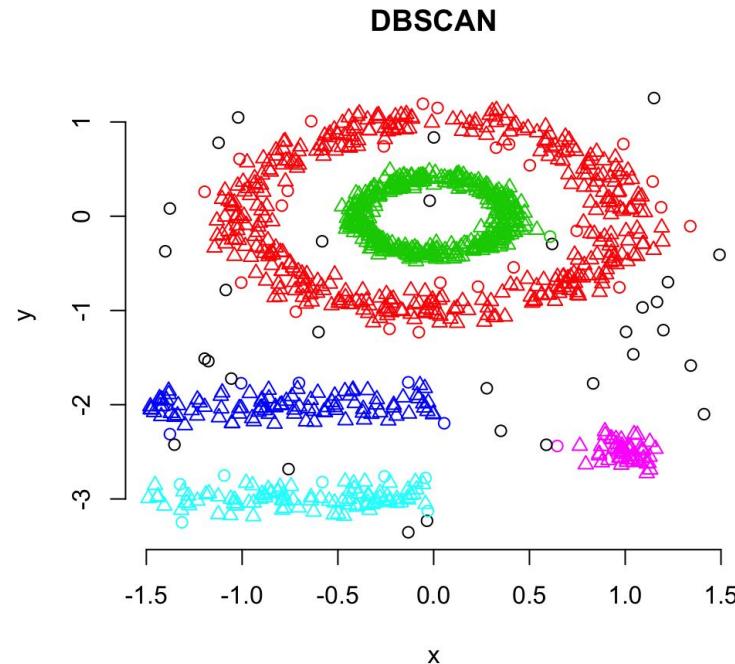
- ◇ Resumindo, o agrupamento de objetos a partir de qualquer cluster de C é um processo de duas etapas.
- ◇ Na primeira, um objeto central arbitrário X do cluster 1 (XC1) é identificado.
- ◇ Em seguida, todos os objetos alcançáveis por densidade a partir de XC1 são buscados.
- ◇ Na segunda etapa, cada cadeia de objetos partindo de XC1 é detectada de forma recursiva.



Exemplos



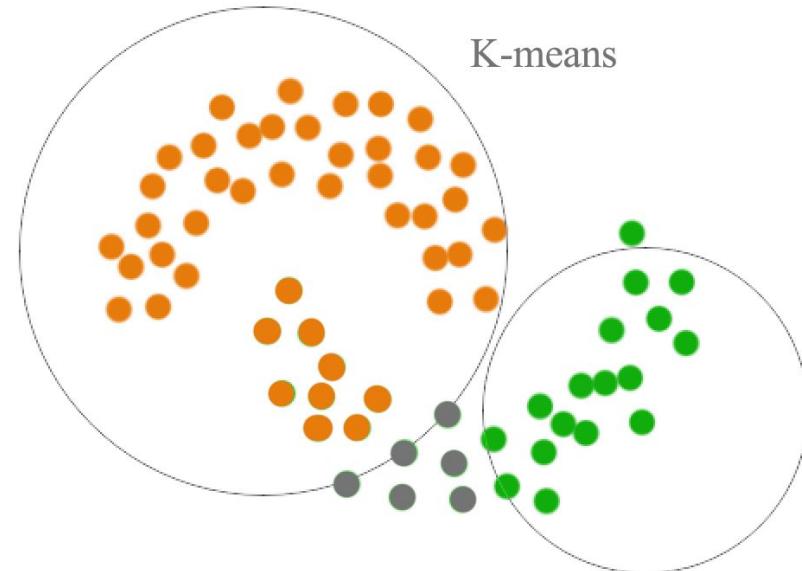
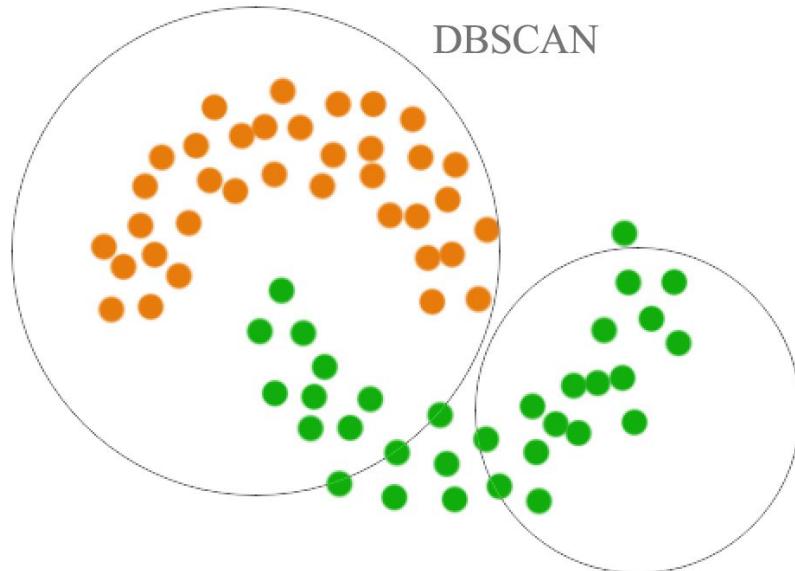
Exemplos





DBSCAN

DBSCAN vs K-Means

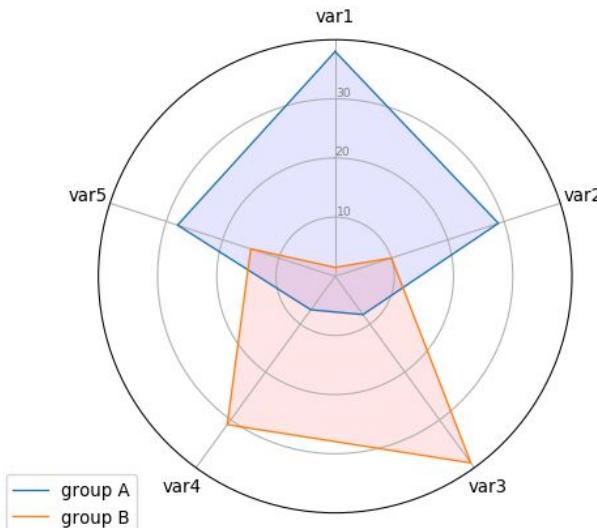




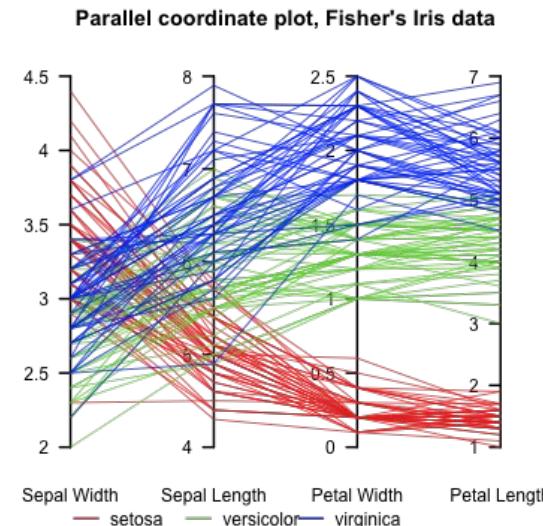
Como definir os hiperparâmetros

- ◊ Depende ...
- ◊ Um minPts baixo significa que ele criará mais clusters, inclusive a partir dos nós que deveriam ser ruído, logo, um valor muito pequeno não deve ser setado.
- ◊ Um raio grande juntamente com um baixo valor de minPts formará poucos clusters.
- ◊ Um raio pequeno com um baixo valor de minPts gerará muitos clusters
- ◊ Papers ...

Visualização



Radar chart



Parallel Coordinates



Hands-On





Deep Learning





“Deep learning é apenas uma forma complexa de uma rede neural”

Booz Allen Hamilton



“Em vez de organizar os dados para serem executados através de equações pré-definidas, o *deep learning* configura parâmetros básicos sobre os dados e treina o computador para aprender sozinho através do reconhecimento padrões em várias camadas de processamento.”



Introdução

- ◆ Tem ajudado os computadores a compreenderem, tem aprimorado o aprendizado de reconhecer e classificar;
- ◆ Novas abordagens têm possibilitado uma maior precisão dos modelos
- ◆ Disponibilidade de um alto volume de dados para treinamento de aprendizagem profunda (e.g.: IOT)



2017 This Is What Happens In An Internet Minute



2018 This Is What Happens In An Internet Minute





Introdução

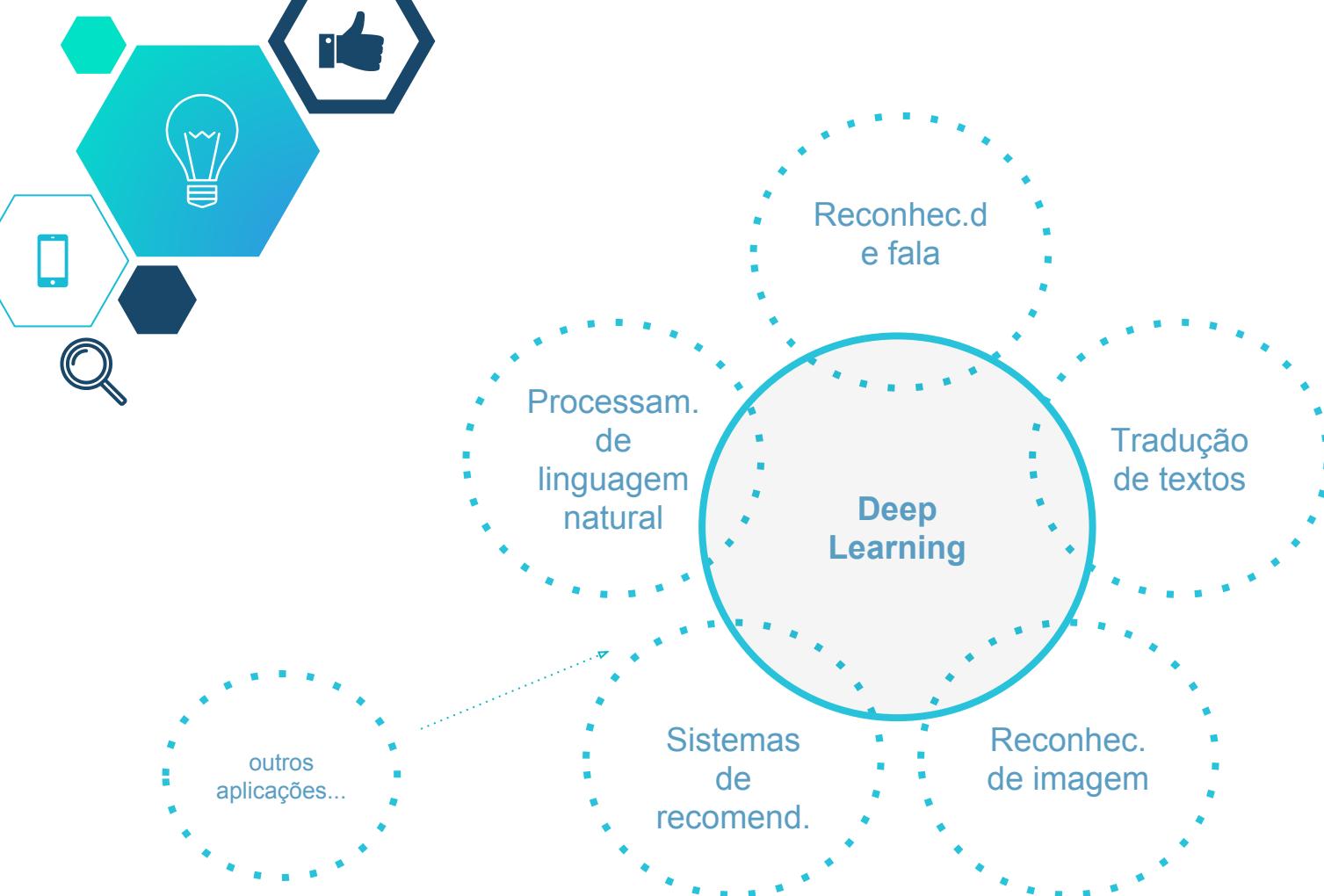
- ◆ Possui a capacidade de se adaptar a mudanças no padrão introduzindo um comportamento mais dinâmico
- ◆ Melhora a precisão e a performance das aplicações que utilizam redes neurais



Introdução

- ◊ Novas classes de redes neurais estão sendo desenvolvidas (e.g.: classificação de imagens e de texto)
- ◊ Os periféricos tem sido trocado por uma interação mais humana, como gestos e fala.
- ◊ Necessidade de poder computacional devido a complexidade da solução que cresce conforme o #camadas aumenta e o grande volume de dados necessário para treinar as redes.

Deep Learning



Introdução

- ◊ Google assistent
- ◊ Google draw
- ◊ Carros autônomos
- ◊ Google translator
- ◊ AlphaGo
- ◊ ...



DL vs ML

- ◊ A maioria dos modelos que usamos precisam de feature engineering
- ◊ A performance do modelo em si depende suas características e o processo de FE é custoso e é específico para cada domínio
- ◊ Cada vez que novas features são adicionadas aos dados é necessário refazer todo o processo de feature engineering e EDA (exploratory data analysis)

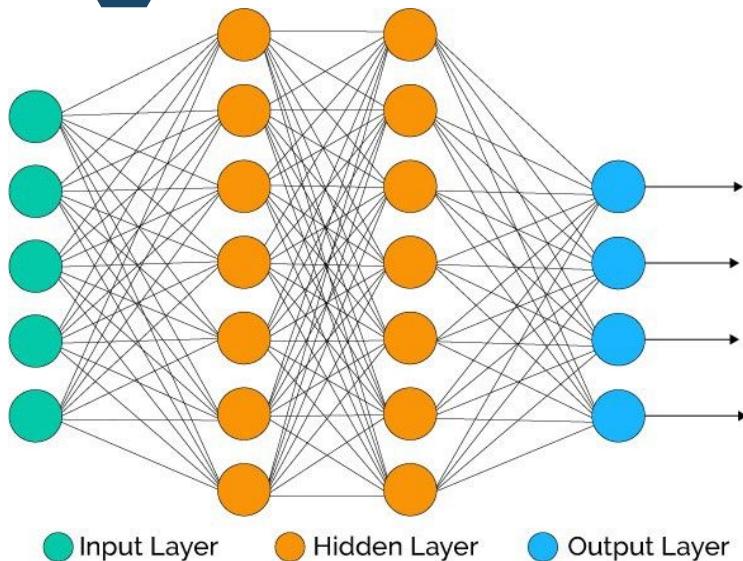


DL vs ML

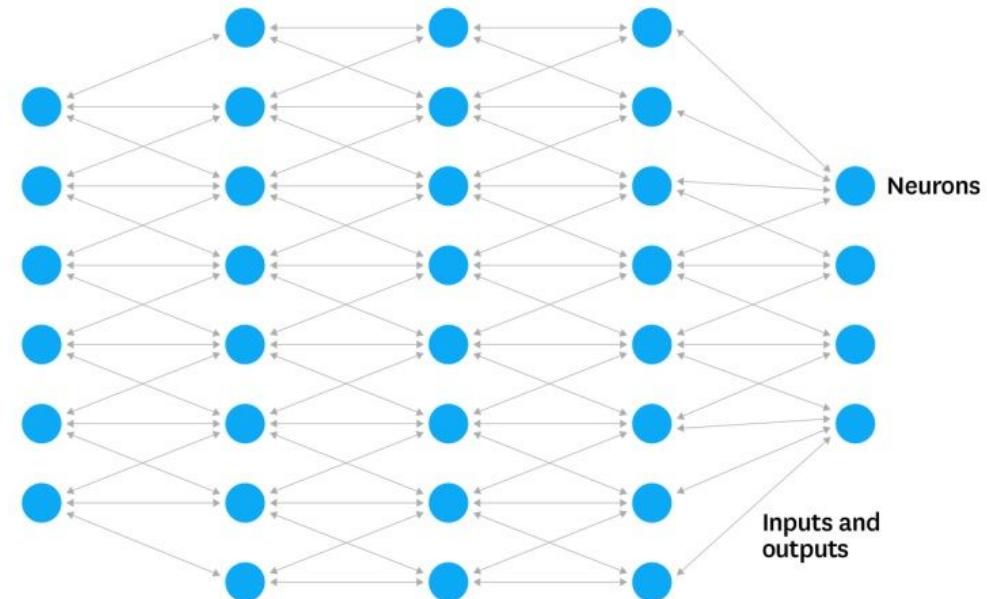
- ◆ DL permite substituir a formulação e a especificação de um modelo por camadas que aprendem a reconhecer as características dos dados em suas camadas.
- ◆ Dispensam grande parte do pré-processamento e geram automaticamente propriedades invariantes em suas camadas hierárquicas de representação.
- ◆ Redes de DL se adaptam melhor e melhoram continuamente à medida que novos dados são adicionados, são mais dinâmicas do que modelos preditivos baseados em regras de negócios



Rede Neural vs DL



Rede Neural

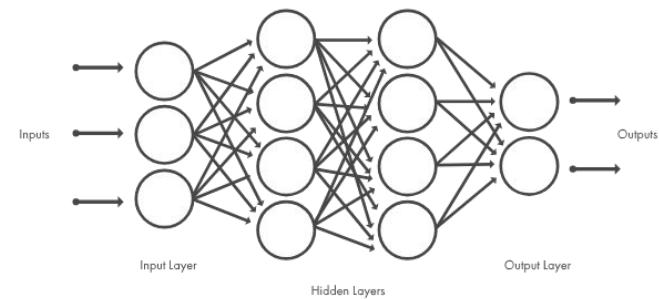


Deep Learning



Redes Neurais Convolucionais

Convnets ou CNN



CNN

- ◊ Algoritmos baseados em redes neurais artificiais que utilizam a convolução em pelo menos uma de suas camadas
- ◊ Provaram ser eficientes em diversas tarefas de reconhecimento de imagens e vídeos, sistemas de recomendação e processamento de linguagem natural
- ◊ Necessitam de uma grande quantidade de amostras rotuladas para o aprendizado

CNN

- ◆ Se tornaram o novo padrão em visão computacional e são fáceis de treinar quando existe grande quantidade de amostras
- ◆ Capacidade de extrair características relevantes através de aprendizado de transformações (kernels)
- ◆ Dependem de um menor número de parâmetros de ajustes do que redes totalmente conectadas com o mesmo número de camadas ocultas

CNN

- ◆ Como cada unidade de uma camada não é conectada com todas as unidades da camada seguinte, há menos pesos para serem atualizados, facilitando assim o treinamento



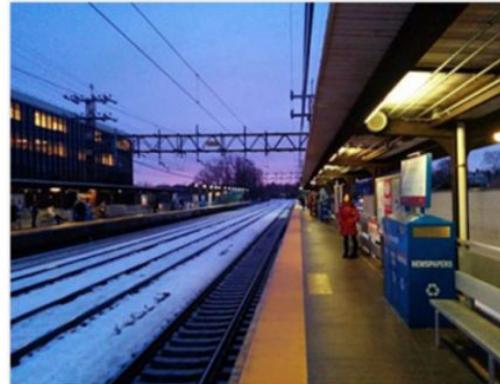
Exemplo

CNN



noite ponte cidade ponte suspensa

rio



trem metrô ferrovia via férrea

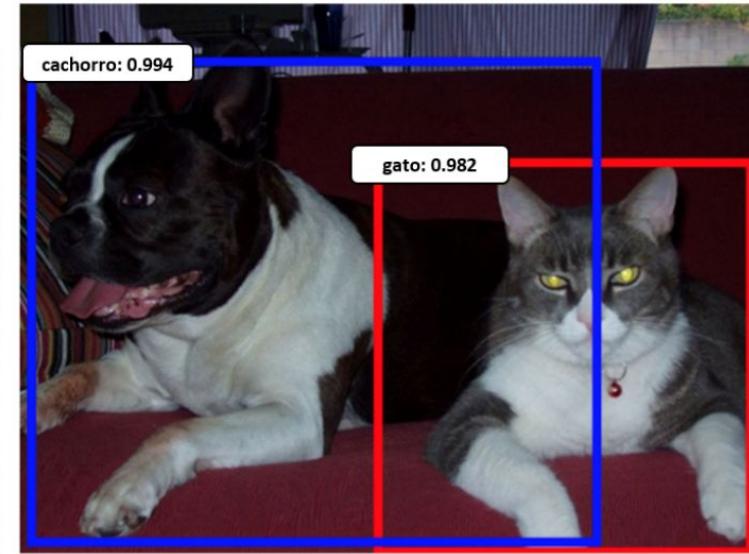
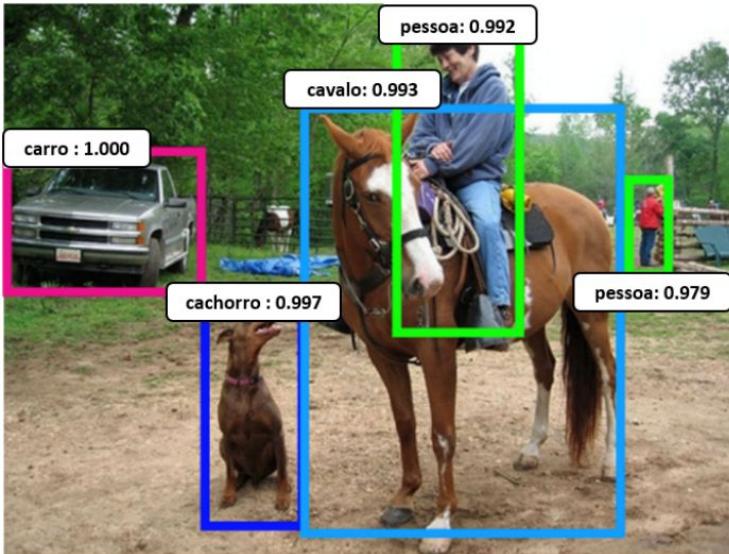
estação transporte



competição trem tênis estádio bola

multidão espectadores

CNN



CNN utilizada para reconhecimento de objetos e segmentação de uma imagem



Histórico

- ◆ 1988 : Foi criada a LeNet, um dos primeiros projetos de CNN, ela foi utilizada para reconhecimento de caracteres (e.g.: dígitos numéricos)
- ◆ 1990 a 2012: Durante este período de, as CNNs estavam em um período de incubação
- ◆ Com o aumento da quantidade de dados disponíveis e do poder computacional, por exemplo, através do uso de GPUs, as CNNs se tornaram cada vez mais eficientes



Histórico

- ◆ 2012: Criada a AlexNet que consiste numa versão mais profunda da LeNet, ou seja, com mais camadas. Essa rede possui 5 camadas convolucionais, camadas de max-pooling e três camadas totalmente conectadas com dropout
 - A rede ganhou o desafio de classificação ILSVRC (ImageNet Large Scale Visual Recognition Challenge)
 - Foi utilizada para classificar imagens em 1000 possíveis categorias.

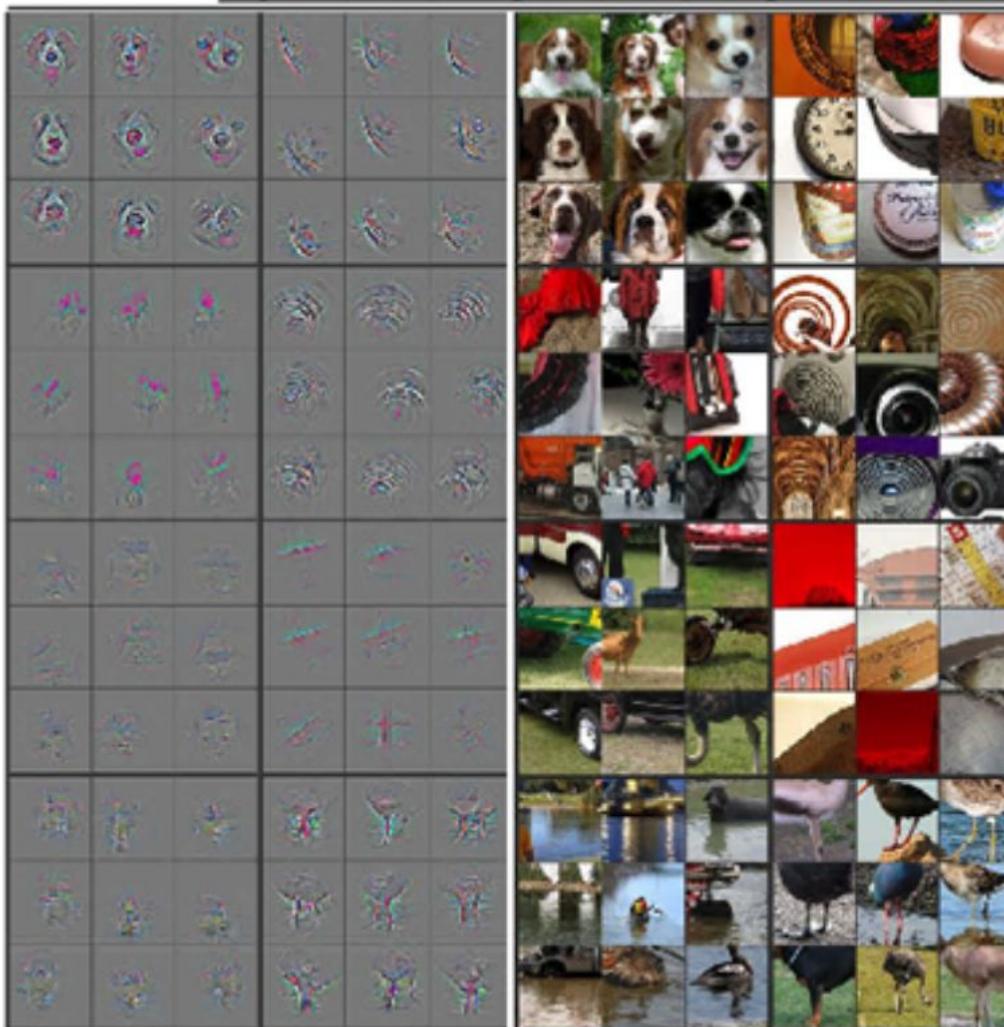


Histórico

- ◊ 2013: Com o sucesso da AlexNet em 2012, foram submetidos diversos modelos baseados em CNN para o ILSVRC 2013, dentre eles a ZF Net (2013)
 - A arquitetura da rede ZF Net consistiu de algumas modificações da AlexNet e diminuição do tamanho dos filtros e do passo na primeira camadas
 - Introdução a técnica DeConvNet que realiza operações reversas de pooling até que o tamanho da imagem ser atingido



DeConvNet



CNN



Histórico

- ◆ 2014: Foi criada a GoogLeNet, essa rede ganhou o ILSVRC 2014 com top-5 erro de 6,67%. Foi o primeiro modelo que introduziu a ideia de que as camadas das CNNs não precisavam ser executadas sequencialmente.
- ◆ 2015: Microsoft e ResNet (2015) vencem o ILSVRC 2015 com top-5 erro de 3.6%. O desempenho dessa rede foi superior ao de seres humanos, que dependendo de suas habilidades e área de conhecimento e normalmente obtém o top-5, erro entre 5 e 10%.



Conceitos

- ◆ CNNs são formadas por sequências de camadas e cada uma destas possui uma função específica na propagação do sinal de entrada
- ◆ As principais suas três principais camadas são: convolucionais, de pooling e totalmente conectadas.

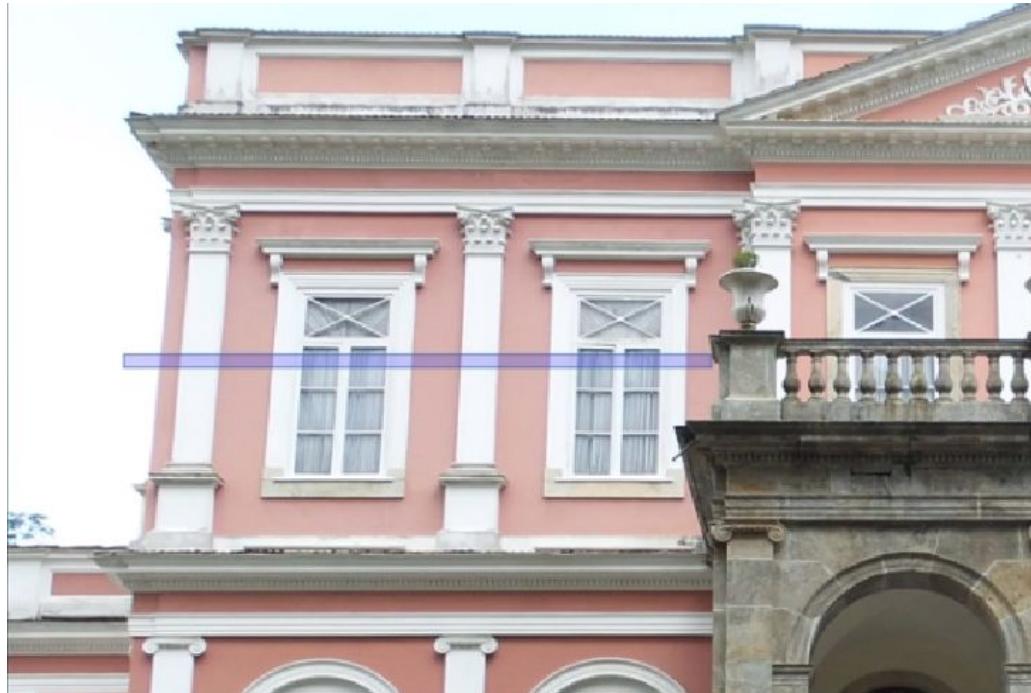


Convolução

- ◆ São responsáveis por extrair atributos dos volumes de entradas
- ◆ Consistem de um conjunto de filtros que geralmente recebem como entrada um arranjo 3D, também chamado de volume
- ◆ Cada filtro possui dimensão reduzida, porém ele se estende por toda a profundidade do volume de entrada
- ◆ Automaticamente, durante o processo de treinamento da rede, esses filtros são ajustados para que sejam ativados em presença de características relevantes identificadas no volume de entrada, como orientação de bordas ou manchas de cores



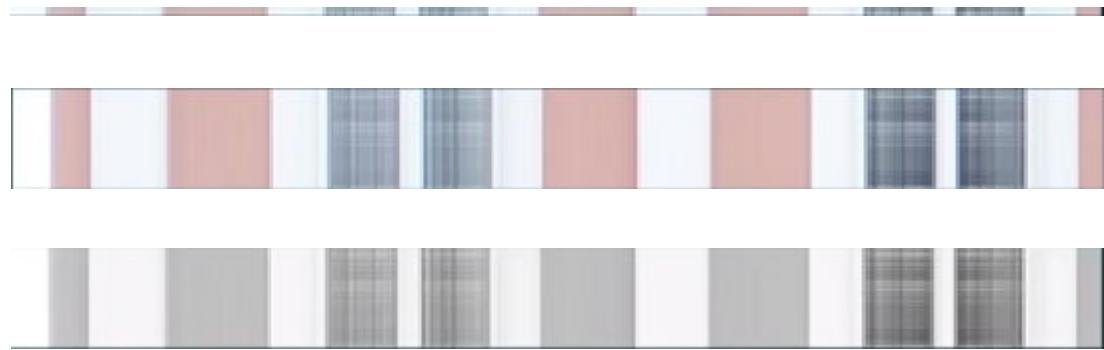
Convolução





CNN

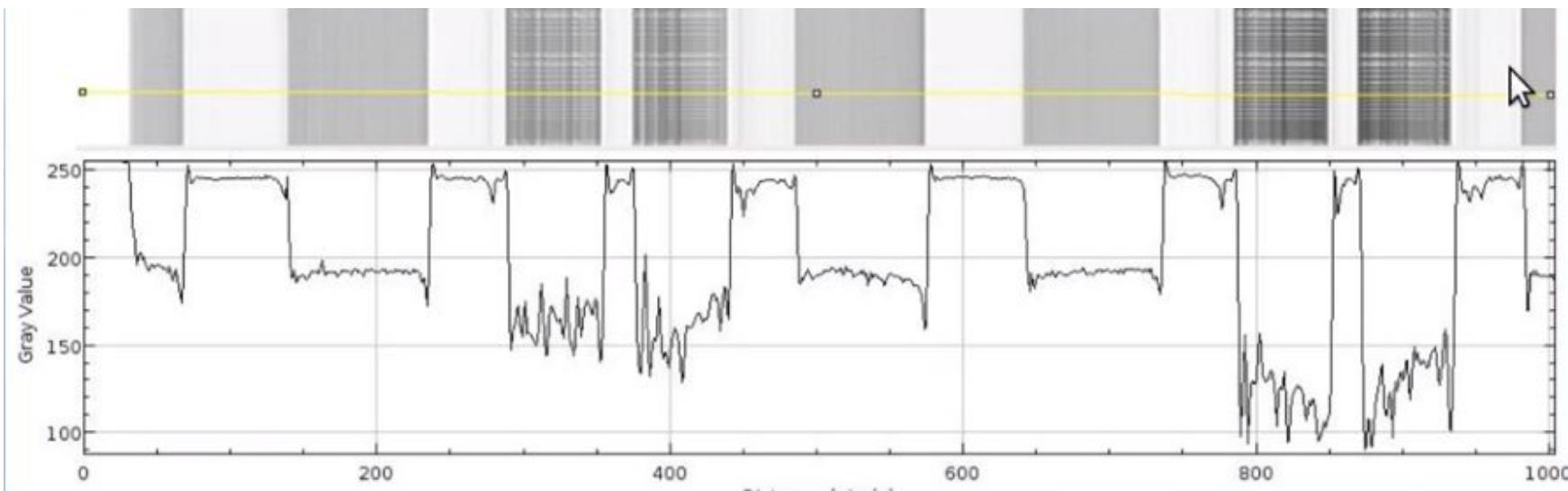
Convolução





CNN

Convolução

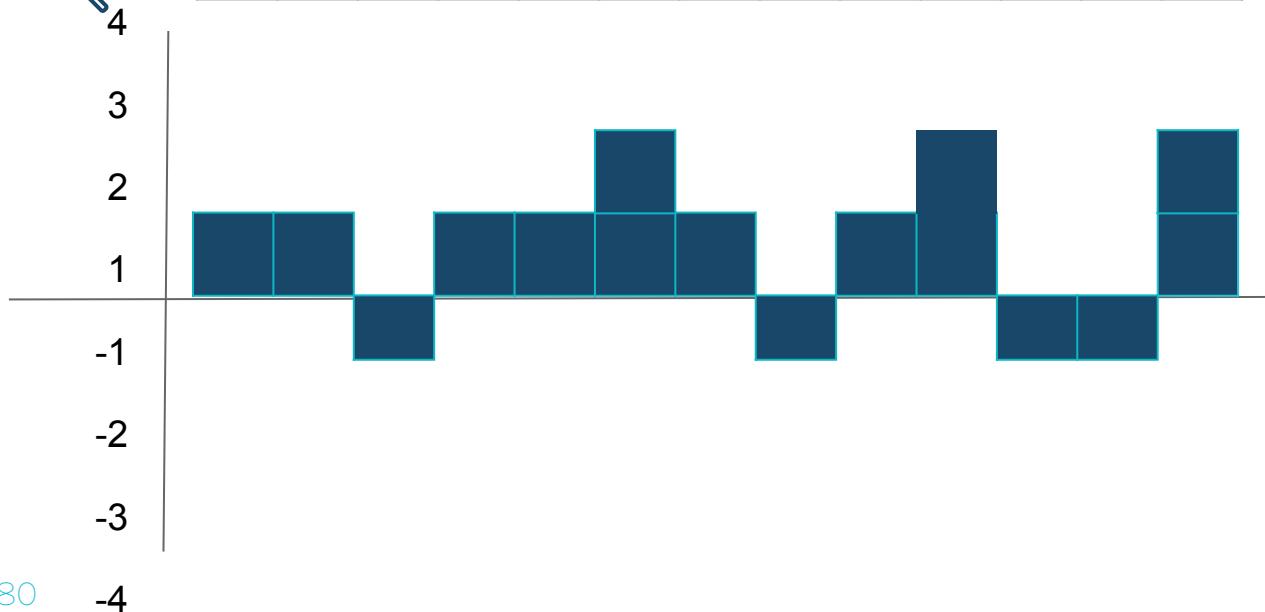




Convolução

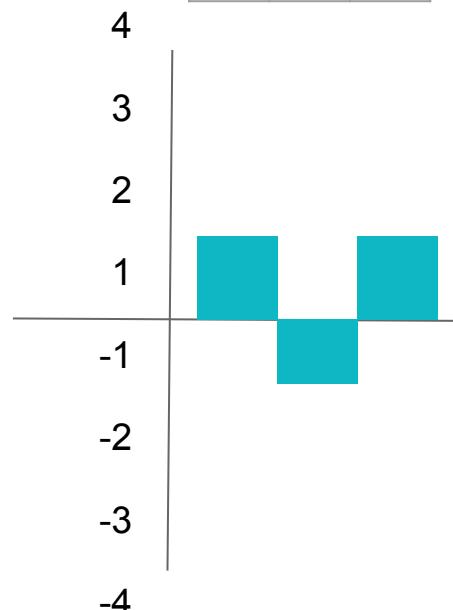
Volume

1	1	-1	1	1	2	1	-1	1	2	-1	-1	2
---	---	----	---	---	---	---	----	---	---	----	----	---



Kernel (núcleo)

1	-1	1
---	----	---

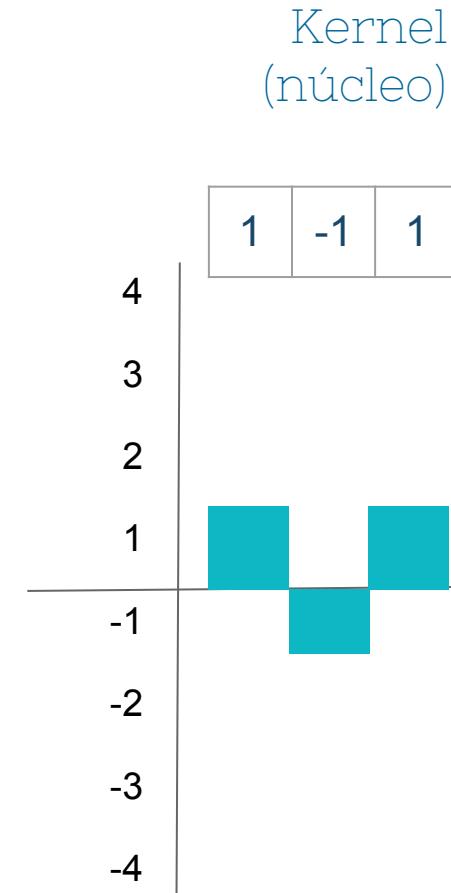
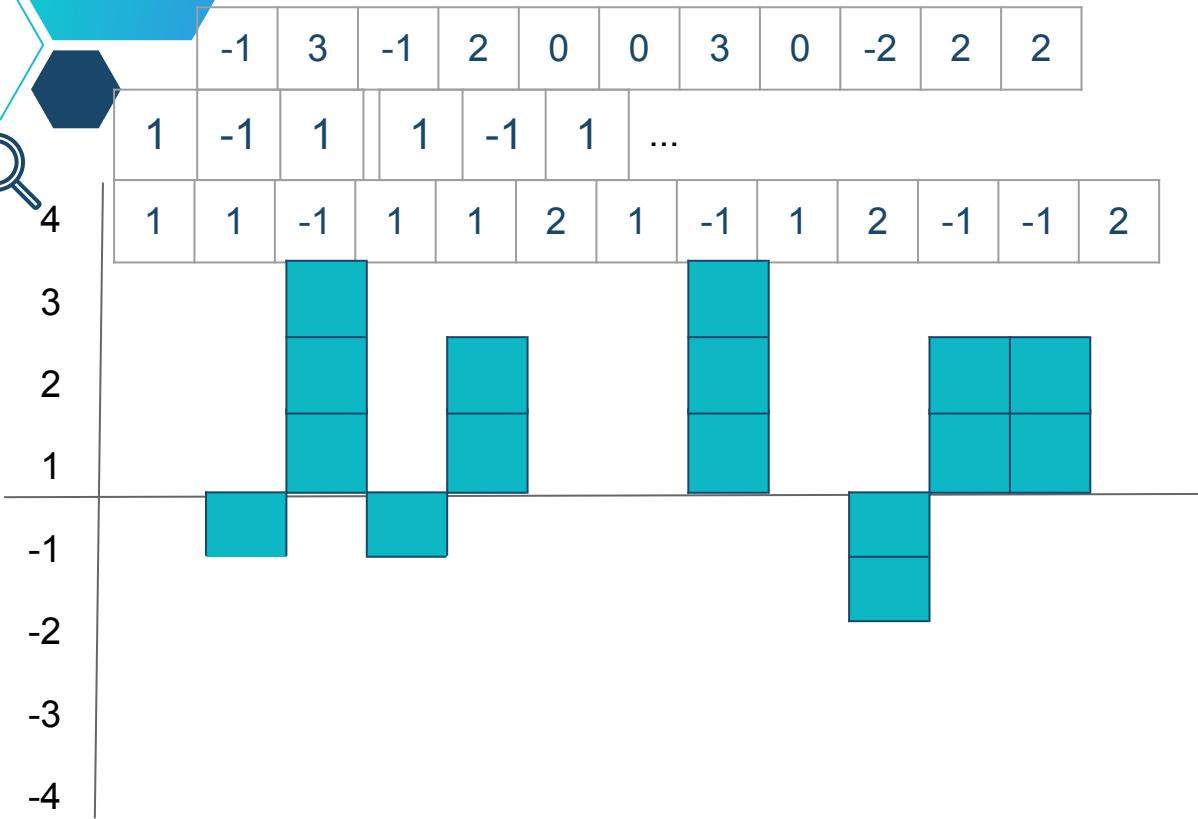




“A somatória do produto ponto a ponto entre os valores de um filtro e cada posição do volume de entrada é uma operação conhecida como convolução”

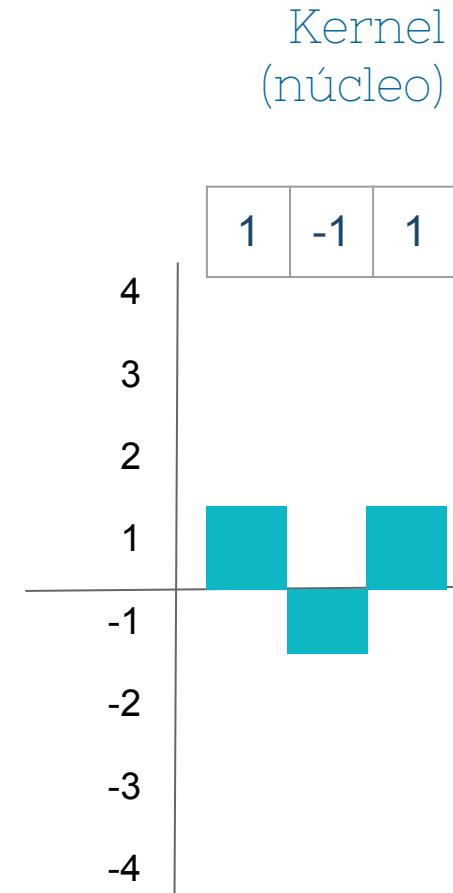
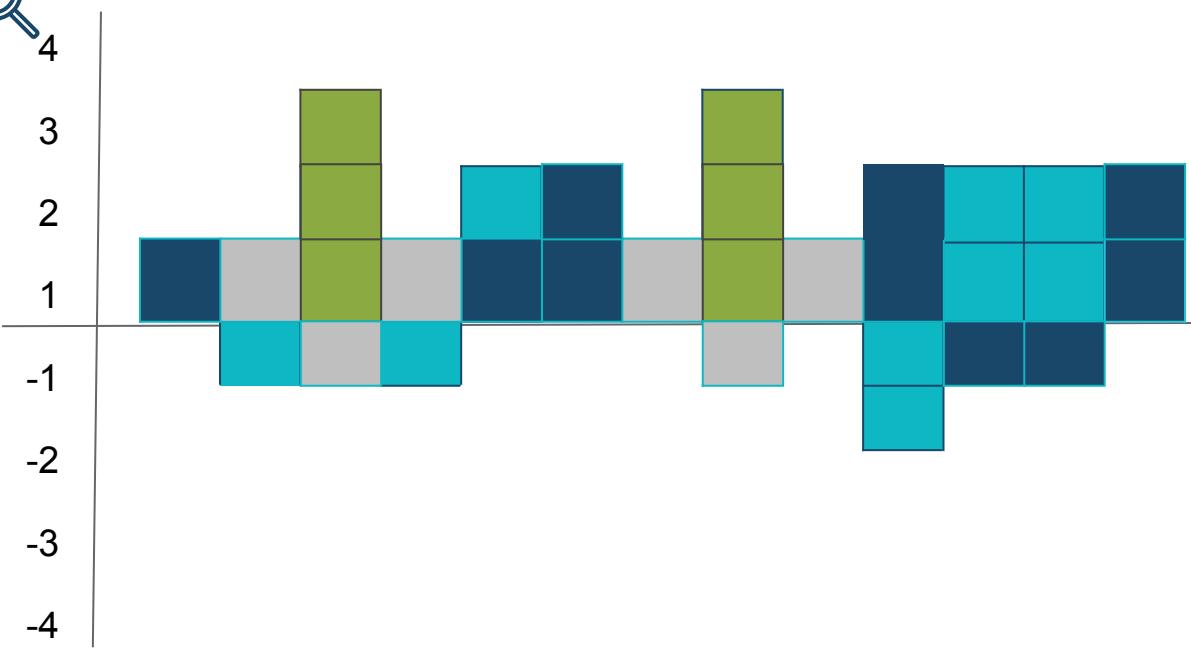


Convolução





Convolução





Convolução

1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	0	0
0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	0
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	1
0	0	1	1	0
0	1	1	0	0

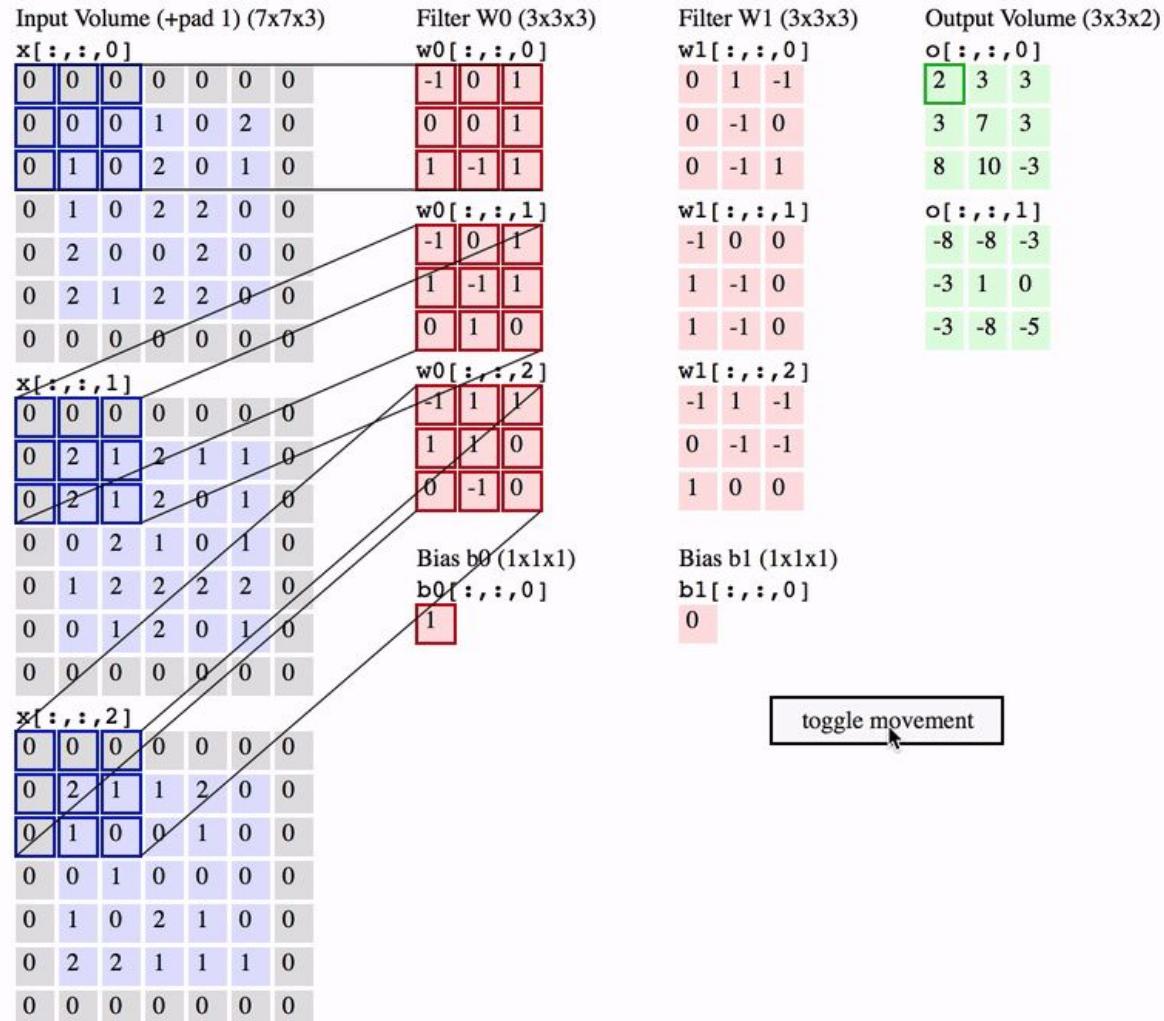
Image

4		

Convolved
Feature

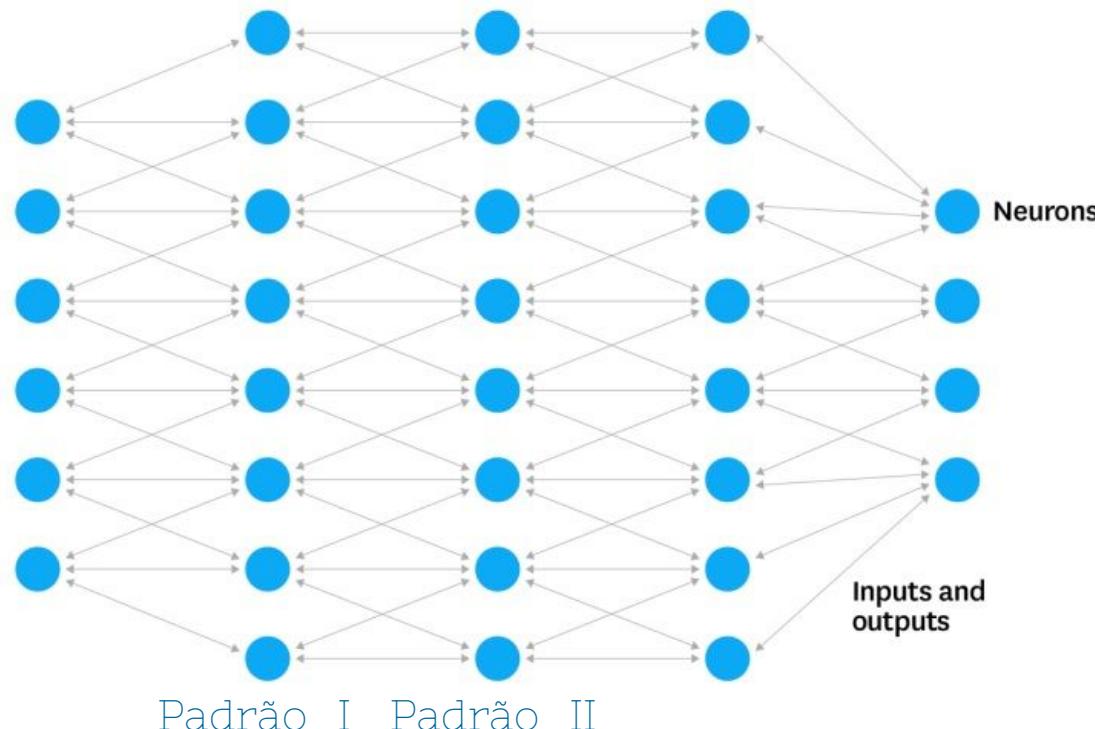


Convolução Imagem Colorida (3D)



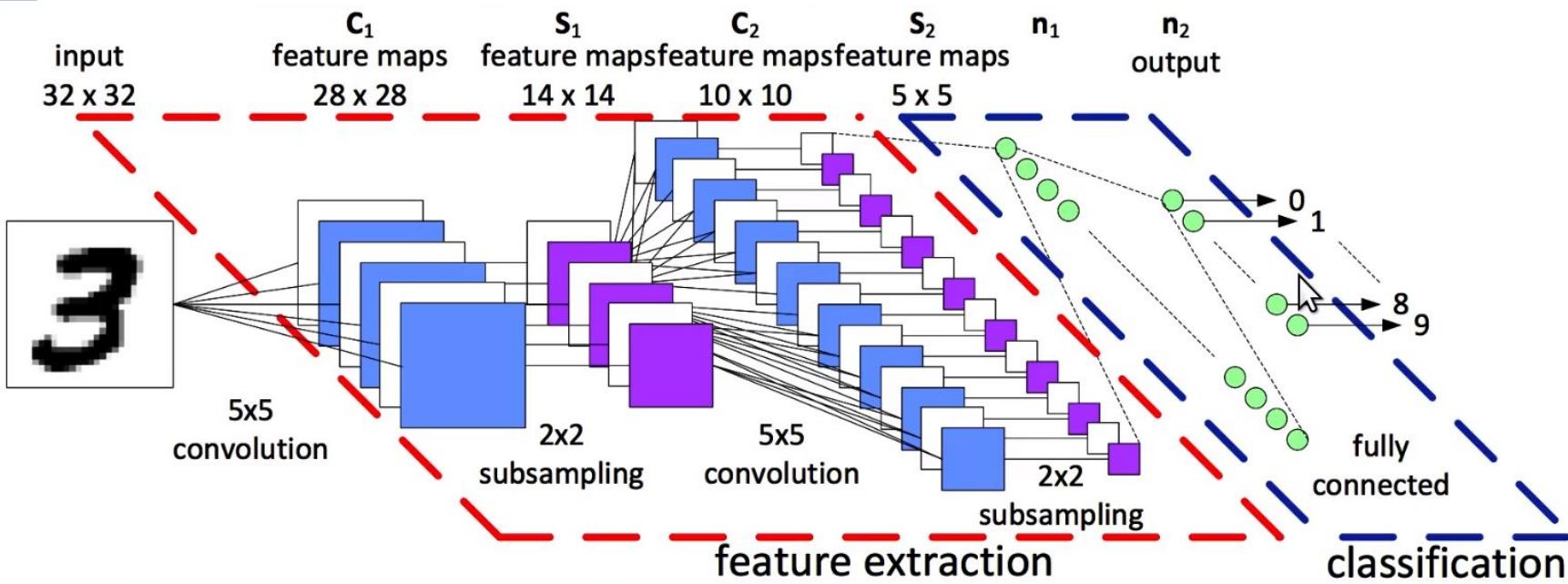


Convolução





Feature extraction





Convolução

- ◆ Existem três parâmetros que controlam o tamanho do volume resultante da camada convolucional: zero-padding, profundidade (depth) e passo (stride)
- ◆ Quanto maior o número de filtros maior o número de características extraídas
- ◆ Quanto maior o #filtros maior a complexidade computacional, relativa ao tempo e ao uso de memória



Passo

- ◆ A altura e largura da matriz resultante dependem do passo e do **zero-padding**
- ◆ O parâmetro passo especifica o tamanho do salto na operação da convolução

Passo = 1

a)

0	0	0	0	0	
0	1	1	1	1	
0	1	2	2	1	
0	1	2	2	2	
0	0	1	2	2	

b)

0	0	0	0	0	
0	1	1	1	1	
0	1	2	2	1	
0	1	2	2	2	
0	0	1	2	2	

Passo = 2

c)

0	0	0	0	0	
0	1	1	1	1	
0	1	2	2	1	
0	1	2	2	2	
0	0	1	2	2	

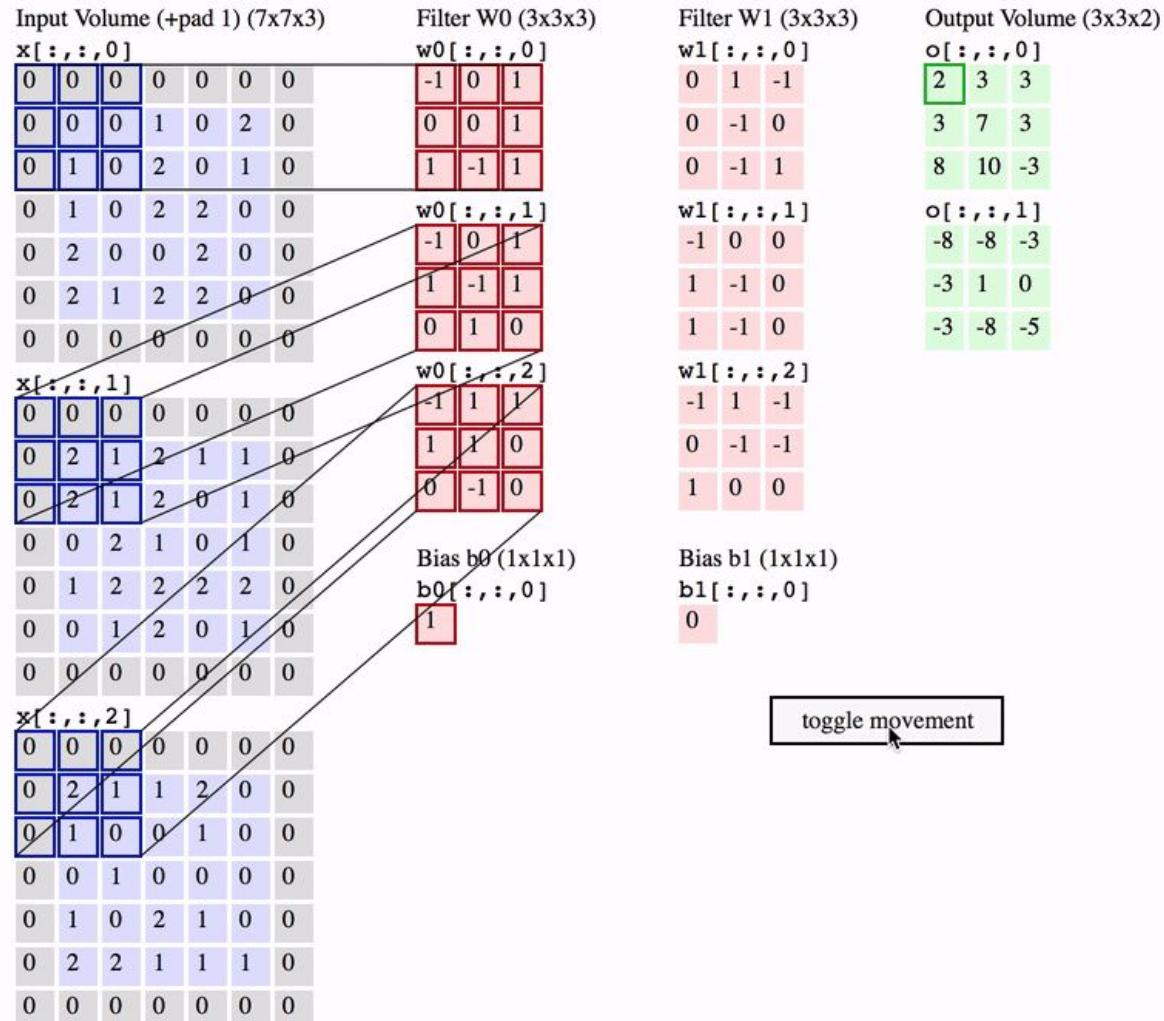
d)

0	0	0	0	0	
0	1	1	1	1	
0	1	2	2	1	
0	1	2	2	2	
0	0	1	2	2	



Convolução Imagem Colorida (3D)

190





Camada de Pooling

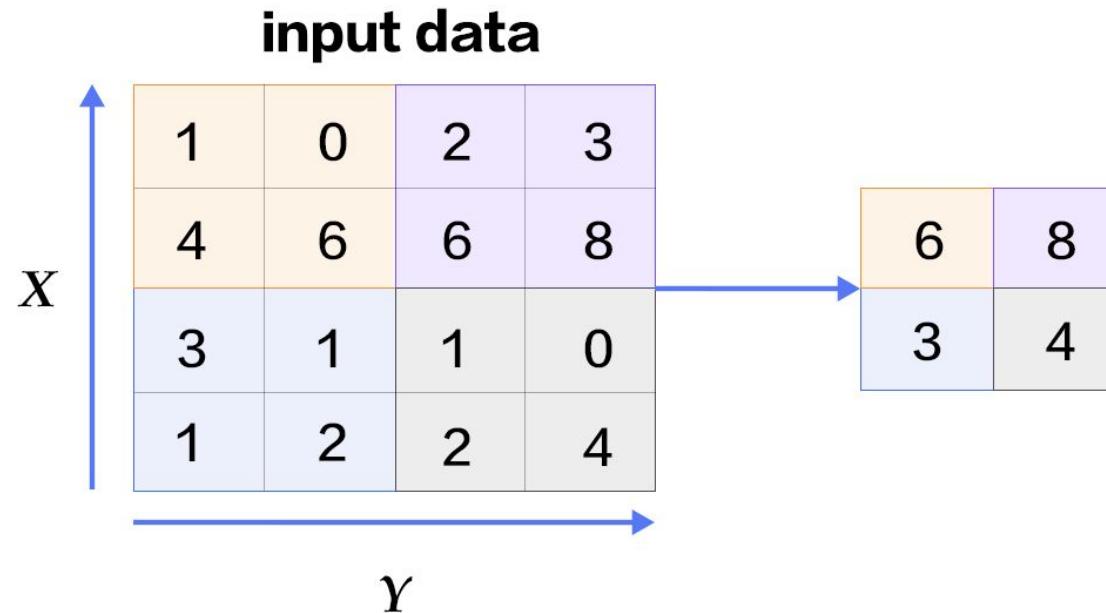
- ◆ Após uma camada convolucional, geralmente existe uma camada de pooling
- ◆ Tem por objetivo reduzir a dimensão espacial do volume de entrada progressivamente
- ◆ Diminui o custo computacional da rede e evita overfitting
- ◆ Na operação de pooling, os valores pertencentes a uma determinada região do mapa de atributos, são substituídos por alguma métrica dessa região



Camada de Pooling

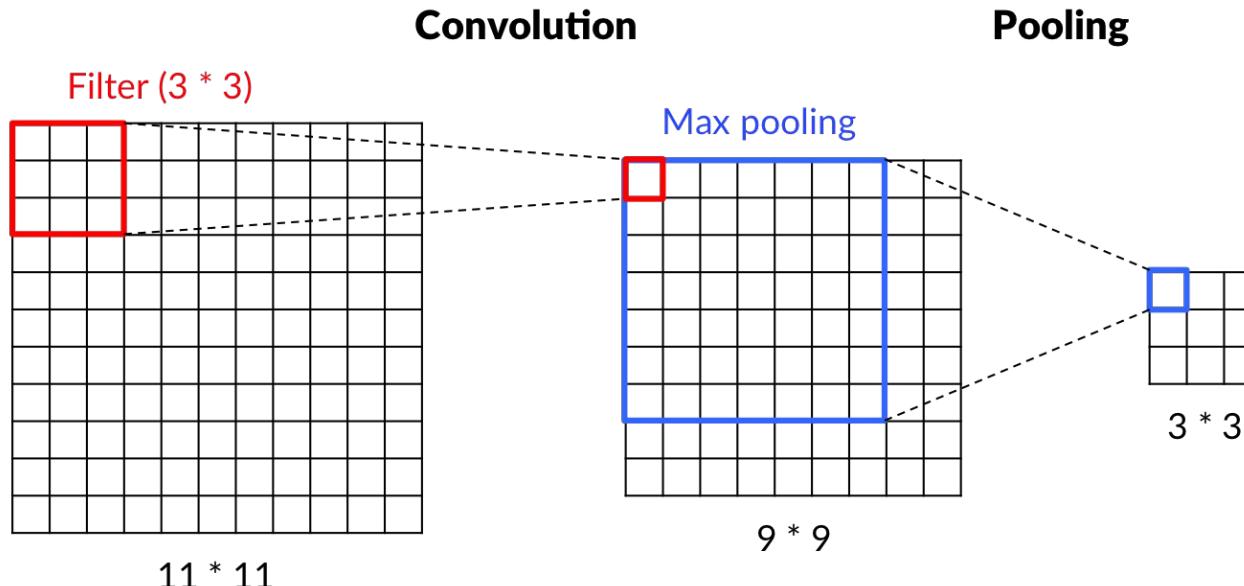
- ◆ A forma mais comum de pooling consiste em substituir os valores de uma região pelo valor máximo
- ◆ Max pooling é útil para eliminar valores desprezíveis, reduzindo a dimensão da representação dos dados e acelerando a computação necessária para as próximas camadas
- ◆ Cria uma invariância a pequenas mudanças e distorções locais

Camada de Pooling



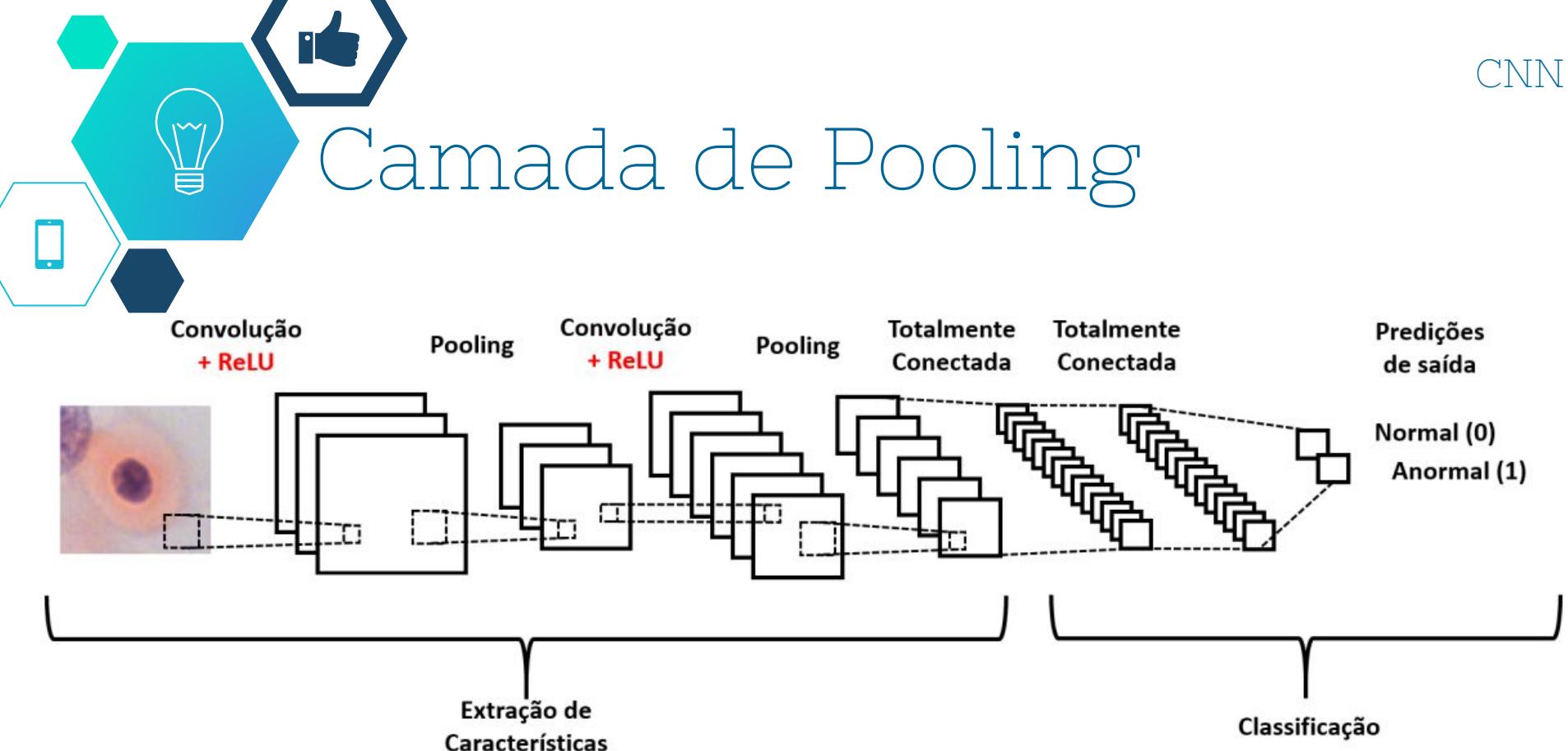
Aplicação de max pooling em uma imagem 4x4 utilizando um filtro 2x2

Camada de Pooling



Exemplo da aplicação da camada de convolução e pooling

Camada de Pooling



Exemplo da extração de features e classificação futura

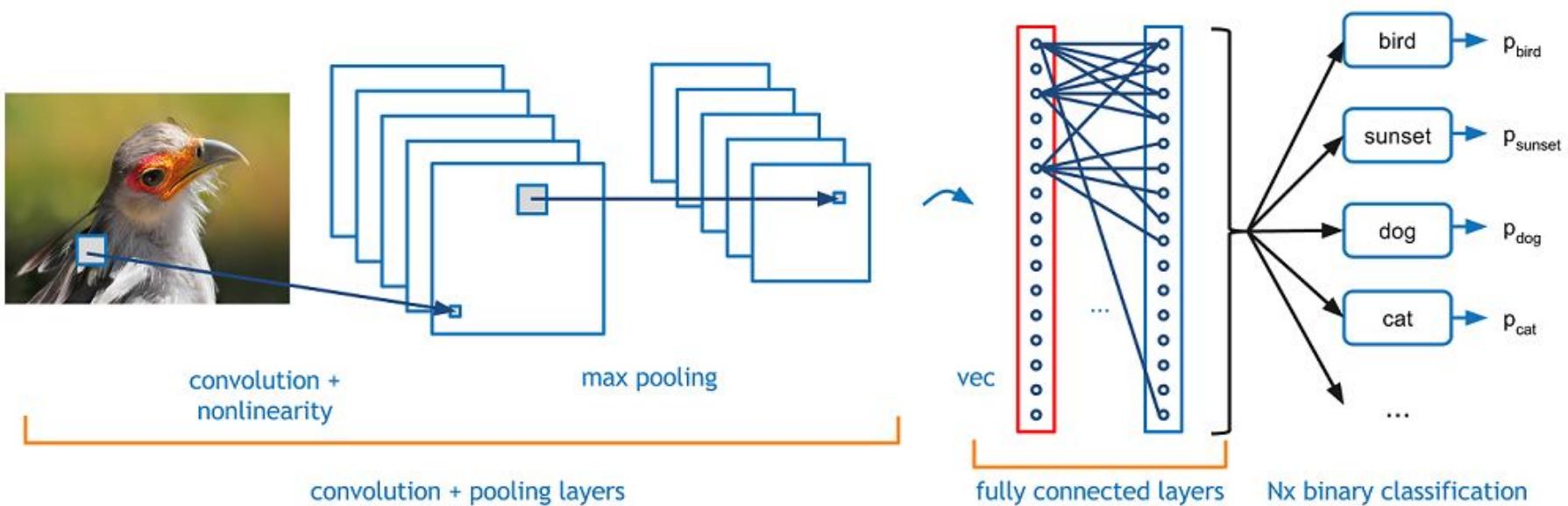


Camada Totalmente conectada

- ◊ A saída das camadas convolucionais e de pooling representam as características extraídas da imagem de entrada
- ◊ O objetivo desta camada é utilizar essas características para realizar a classificação, por exemplo uma imagem, em uma classe pré-determinada.



Camada Totalmente Conectada



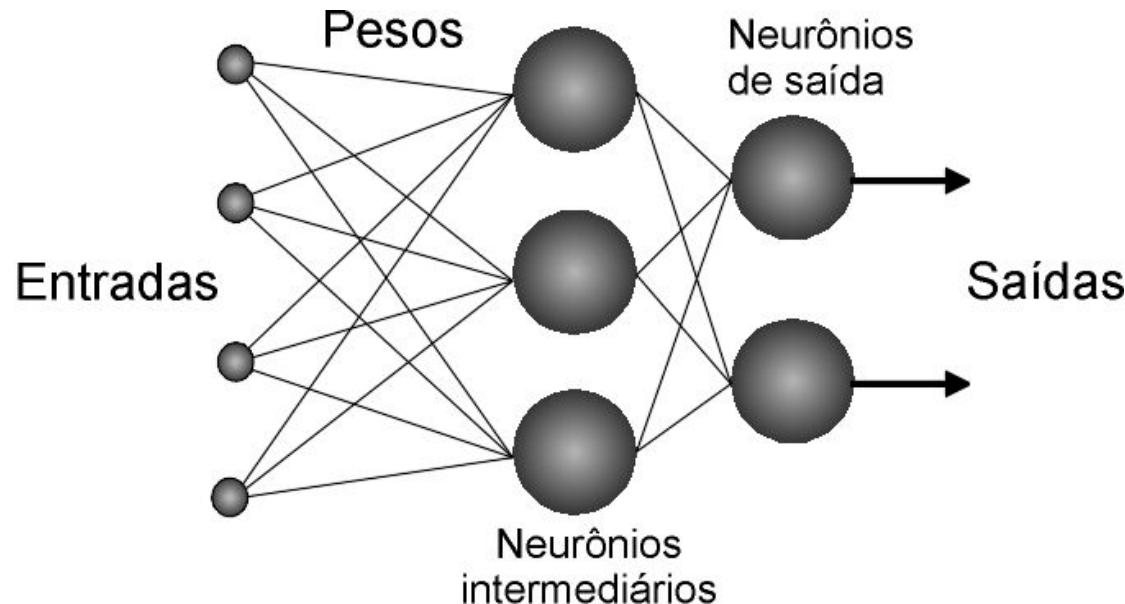


Camada Totalmente conectada

- ◆ As camadas totalmente conectadas são exatamente iguais a uma rede neural artificial convencional (Multi Layer Perceptron ou MLP) que usa a função de ativação **softmax**
- ◆ Essas camadas são formadas por unidades de processamento conhecidas como neurônio, e o termo “totalmente conectado” significa que todos os neurônios da camada anterior estão conectados a todos os neurônios da camada seguinte



Camada Totalmente conectada





Camada Totalmente conectada

- ◆ A função softmax recebe um vetor de valores como entrada e produz a distribuição probabilística da imagem de entrada pertencer a cada uma das classes na qual a rede foi treinada. Vale destacar que a soma de todas as probabilidades é igual a 1.
- ◆ Ela normaliza as probabilidades, por exemplo, se temos as saídas como [1.2, 0.9, 0.75], quando aplicamos a função softmax, obteremos [0.42, 0.31, 0.27].

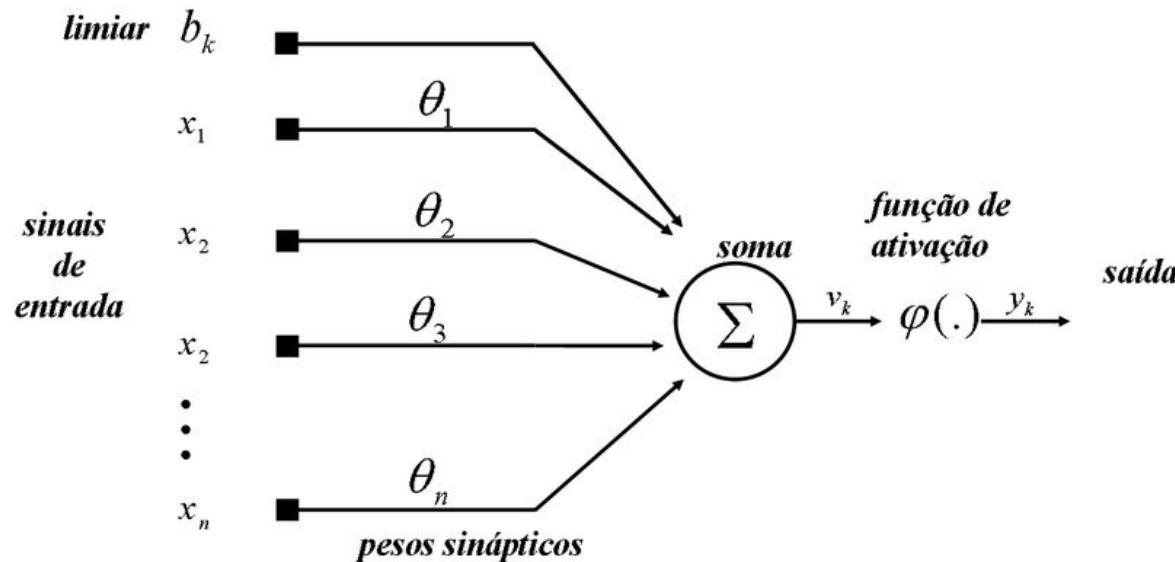


Softmax Function

$$F(X_i) = \frac{\text{Exp}(X_i) \quad i = 0, 1, 2, \dots k}{\sum_{j=0}^k \text{Exp}(X_j)}$$



Função de ativação igual a redes neurais





Camada Totalmente conectada

- ◆ A técnica **dropout** também é bastante utilizada entre as camadas totalmente conectadas para reduzir o tempo de treinamento e evitar overfitting.
- ◆ Consiste em remover, aleatoriamente a cada iteração de treinamento, uma determinada porcentagem dos neurônios de uma camada, readicionando-os na iteração seguinte
- ◆ Confere à rede a habilidade de aprender atributos mais robustos, uma vez que um neurônio não pode depender da presença específica de outros neurônios



Backpropagation

- ◆ A forma mais comum de treinamento de uma CNN é por meio do algoritmo backpropagation
- ◆ **Passo 1:** Todos os filtros e pesos da rede são inicializados com valores aleatórios;
- ◆ **Passo 2:** A rede recebe uma imagem de treino como entrada e realiza o processo de propagação (convoluçãoes, ReLU e pooling, e processo de propagação nas camadas totalmente conectadas). Após esse processo a rede obtém um valor de probabilidade para cada classe.



Backpropagation

- ◊ **Passo 3:** É calculado o erro total obtido na camada de saída (somatório do erro de todas as classes):
$$\text{Erro total} = \sum \frac{1}{2} * (\text{probabilidade_real} - \text{probabilidade_obtida})^2$$
- ◊ **Passo 4:** O algoritmo Backpropagation é utilizado para calcular os valores dos gradientes do erro. Em seguida, a técnica do gradiente descendente é utilizada para ajustar os valores dos filtros e pesos na proporção que eles contribuíram para o erro total. Devido ao ajuste realizado, o erro obtido pela rede é menor a cada vez que uma mesma imagem passa pela rede.
- ◊ **Passo 5:** Repete os passos 2-4 para todas as imagens do conjunto de treinamento.



Considerações

- ◊ Há estratégias específicas para:
 - Tratar as bordas:
 - Zerar, ignorar, replicar, etc.
 - Definir o tamanho do núcleo
 - Definir o #núcleos
 - Definir os núcleos



Considerações

- ◊ Na prática não é comum treinar uma CNN, com pesos aleatórios, geralmente é usado os pesos de uma rede já treinada para uma base muito grande, como a ImageNet que possui mais de 1 milhão de imagens e 1000 classes
- ◊ CNN como Extrator de Características
 - Remover a última camada totalmente conectada da rede (a camada que computa a probabilidade da imagem de entrada pertencer a umas das classes pré-determinadas) e aplicar aprendizagem supervisionada
 - Essa estratégia de extração de características é bastante utilizada para aplicações de imagens médicas, de materiais, etc.



Considerações

- ◆ Você técnicas de transferência de aprendizado para cada situação descrita abaixo:
 - Nova base de imagens é pequena e similar a base original
 - Nova base de imagens é grande e similar a base original
 - Nova base de imagens é pequena e muito diferente da base original
 - Nova base de imagens é grande e muito diferente da base original
 - Maiores informações neste [link](#)

Hands-On

- ◊ Instale o tensorflow via anaconda navigator
- ◊ Veja este [exemplo](#) de como utilizar o tensorflow
- ◊ Aplique o exemplo sobre os dados MNIST (dataset de dígitos manuscritos, 60.000 amostras de treinamento e um conjunto de teste de 10.000 exemplos.)
- ◊ Aplique a solução sobre os dados Fashion MNIST (dataset com produtos de moda de 10 categorias e 6.500 imagens por categoria. O conjunto de treinamento tem 55.000 imagens e o conjunto de teste possui 10.000 imagens.)

Hands-On

- ◊ Caso tenha dificuldades este [código](#) tem uma solução.
- ◊ Faça alterações nos hiperparâmetros da rede, como por exemplo, o número de núcleos, o tamanho dos núcleos, para aumentar a acurácia da rede.
- ◊ Desenhe algum exemplo dependendo da rede treinada, por exemplo um número, tire uma foto, converta para o tamanho correto e aplique a classificação =D.