Aprendizado de Máquina

O Cenário do Aprendizado de Máquina

Prof. Regis Pires Magalhães

regismagalhaes@ufc.br

GÉRON, Aurélien; **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras & TensorFlow:** Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes. 2ª Ed. Alta Books, 2021.

PARTE I - Os conceitos básicos do aprendizado de máquina

1. O Cenário do Aprendizado de Máquina

2. Projeto de Aprendizado de Máquina Ponta a Ponta
3. Classificação
4. Treinando Modelos
5. Máquinas de Vetores de Suporte
6. Árvores de Decisão
7. Aprendizado Ensemble e Florestas Aleatórias (Bagging, Random Forests, Boosting, Stacking)
8. Redução de Dimensionalidade (PCA, Kernel PCA, LLE)
9. Técnicas de Aprendizado Não Supervisionado (Clusterização, Misturas de gaussianas)

PARTE II - Redes Neurais e Aprendizado Profundo

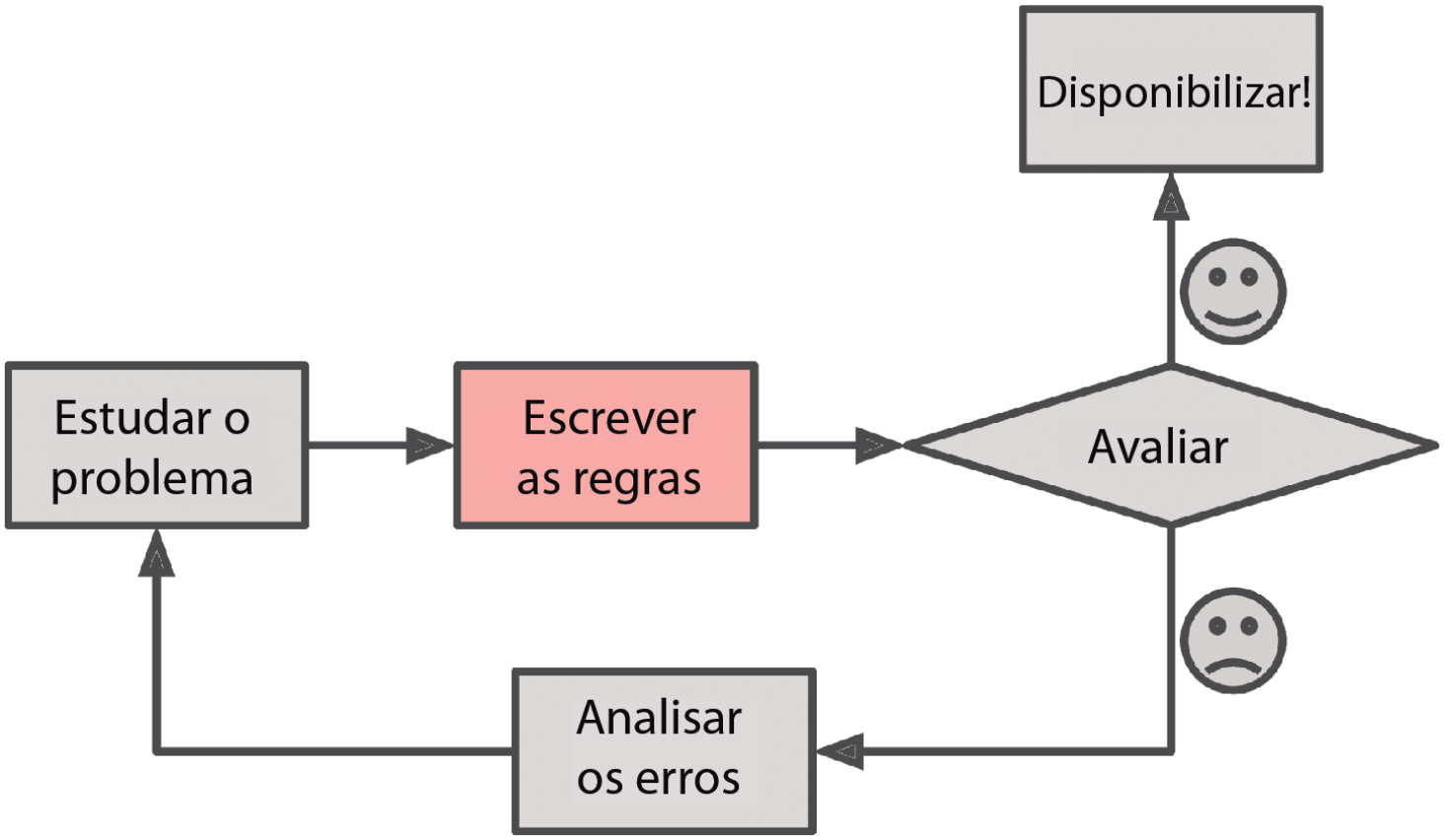
10. Introdução às Redes Neurais Artificiais com a Biblioteca Keras
11. Treinando Redes Neurais Profundas
12. Modelos Customizados e Treinamento com a Biblioteca TensorFlow
13. Carregando e Pré-processando Dados com a TensorFlow
14. Visão Computacional Detalhada das Redes Neurais Convolucionais
15. Processamento de Sequências Usando RNNs e CNNs
16. Processamento de Linguagem Natural com RNNs e Mecanismos de Atenção
17. Aprendizado de Representação e Aprendizado Gerativo com Autoencoders e GANs
18. Aprendizado por Reforço
19. Treinamento e Implementação de Modelos TensorFlow em Larga Escala

O que é Aprendizado de Máquina?

- Aprendizado de máquina é a ciência (e a arte) da programação de computadores de modo que eles possam aprender com os dados.
- [Aprendizado de máquina é o] campo de estudo que possibilita aos computadores a habilidade de aprender sem explicitamente programá-los. — **Arthur Samuel, 1959**
- Alega-se que um programa de computador aprende pela experiência E em relação a algum tipo de tarefa T e alguma medida de desempenho P se o seu desempenho em T, conforme medido por P, melhora com a experiência E. — **Tom Mitchell, 1997**

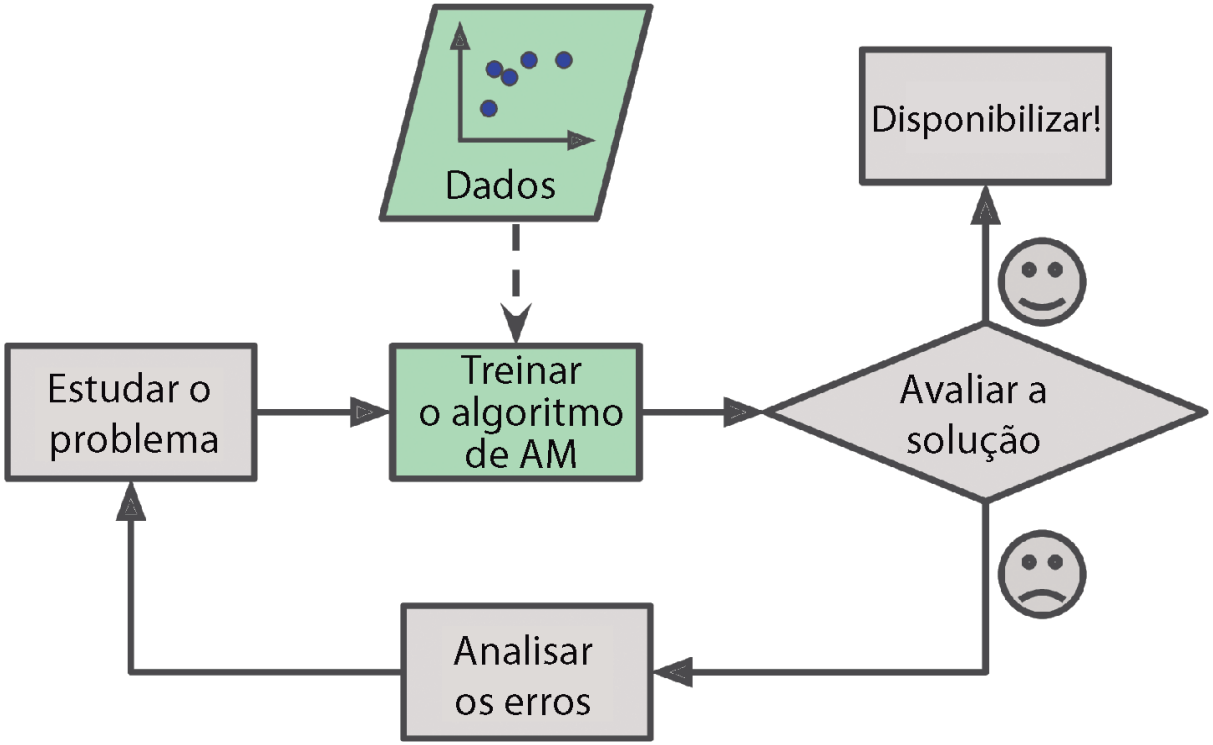
Por que Usar o Aprendizado de Máquina?

Técnicas de programação tradicionais



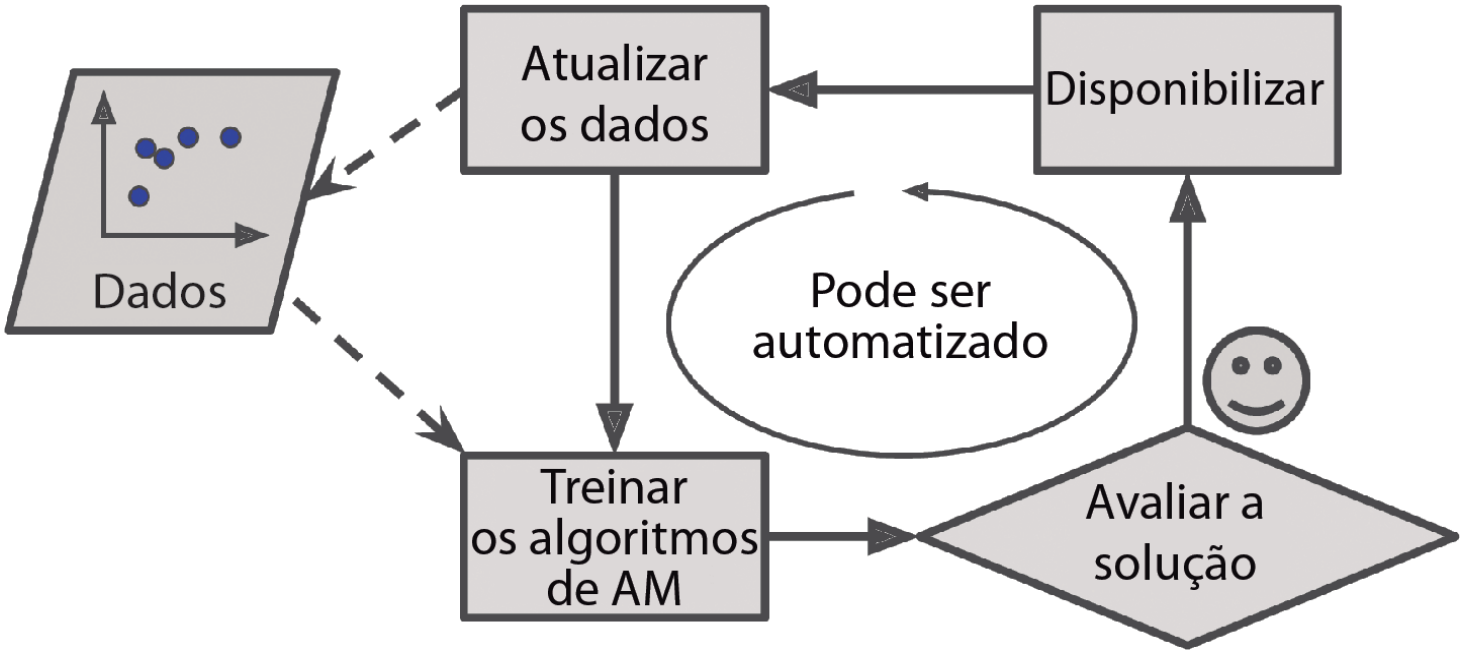
Por que Usar o Aprendizado de Máquina?

Aprendizado de Máquina



Por que Usar o Aprendizado de Máquina?

Adaptando-se automaticamente à mudança

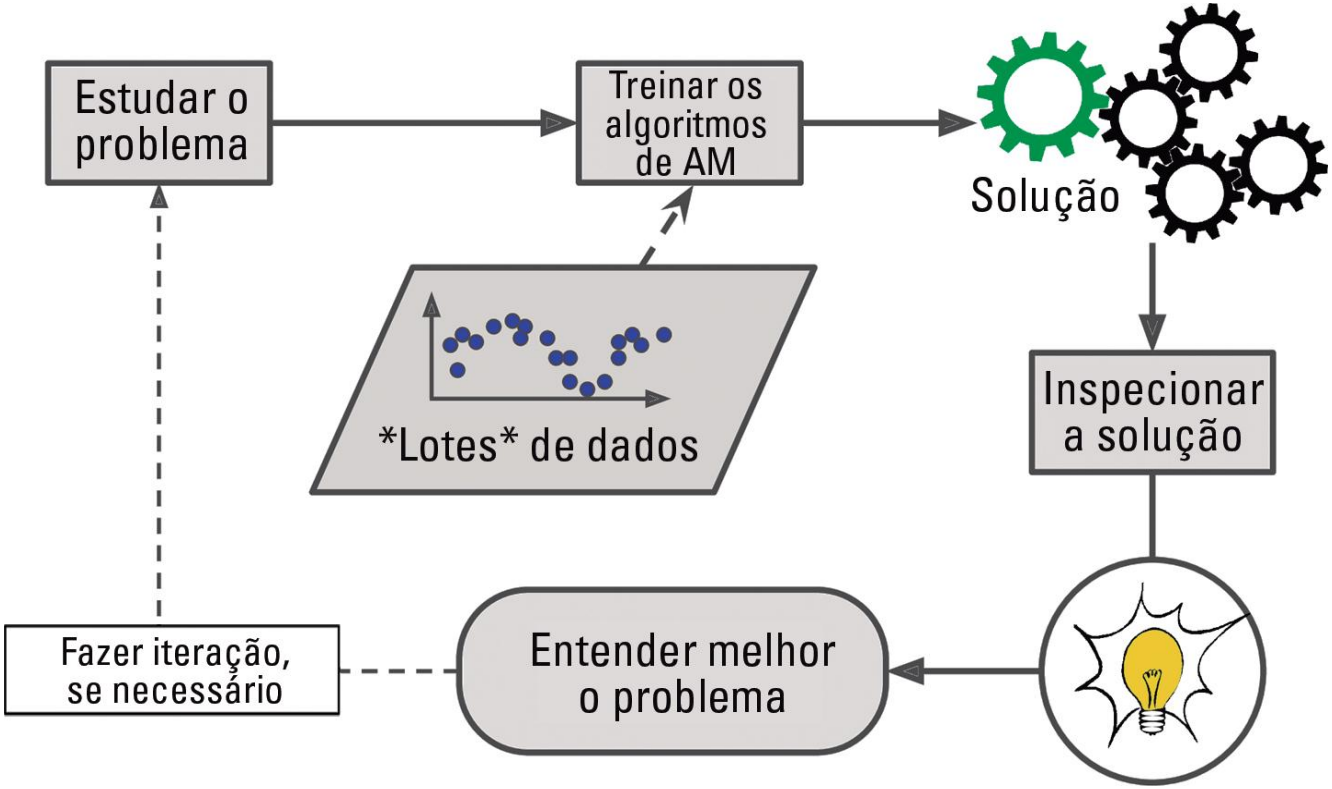


Por que Usar o Aprendizado de Máquina?

- Aplicar técnicas do AM para se aprofundar em grandes quantidades de dados pode ajudar na descoberta de padrões que não eram explícitos. Isso se chama mineração de dados.

Por que Usar o Aprendizado de Máquina?

O aprendizado de máquina pode ajudar no ensino de humanos



Por que Usar o Aprendizado de Máquina?

O aprendizado de máquina é ótimo para:

- Problemas para os quais as soluções atuais exigem muitos ajustes finos ou extensas listas de regras.
- Problemas complexos para os quais não existe uma boa solução quando utilizamos uma abordagem tradicional.
- Adaptabilidade de ambientes: um sistema de aprendizado de máquina pode se adaptar a novos dados.
- Entendimento de problemas complexos e grandes quantidades de dados.

Exemplos de Aplicações

- Análise de imagens de produtos em uma linha de produção a fim de classificá-los automaticamente.
- Detecção de tumores a partir de exames de imagens cerebrais.
- Classificação automática de artigos de notícias.
- Sinalização automática de comentários ofensivos em fóruns de discussão.
- Resumo automático de documentos extensos.
- Criação de um chatbot ou de um assistente pessoal.
- Previsão do faturamento da sua empresa no próximo ano, com base em muitas métricas de desempenho.
- Fazer seu app responder aos comandos de voz.
- Detecção de fraudes com cartão de crédito.
- Segmentação de clientes com base em suas compras, para que você possa elaborar uma estratégia de marketing diferente para cada segmento.

Exemplos de Aplicações

- Representação de um conjunto de dados complexos e de alta dimensão em um diagrama claro e criterioso.
- Recomendação de um produto no qual um cliente possa se interessar, com base em compras anteriores.
- Criar um bot inteligente para um jogo.

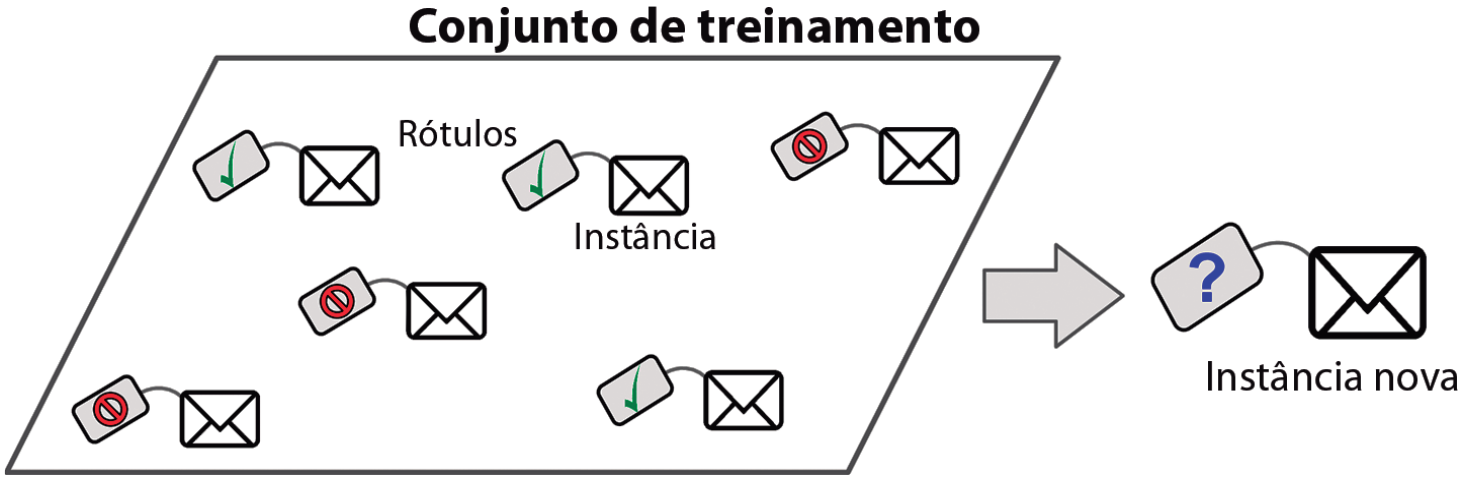
Tipos de Sistemas do Aprendizado de Máquina

- Serem ou não treinados com supervisão humana (aprendizado supervisionado, não supervisionado e semi-supervisionado e aprendizado por reforço).
- Se podem ou não aprender gradativamente em tempo real (aprendizado online vs. aprendizado em batch).
- Se funcionam simplesmente comparando novos pontos de dados com pontos de dados conhecidos, ou se detectam padrões em dados de treinamento e criam um modelo preditivo, como os cientistas (aprendizado baseado em instâncias vs. aprendizado baseado em modelo).

Aprendizado Supervisionado / Não Supervisionado

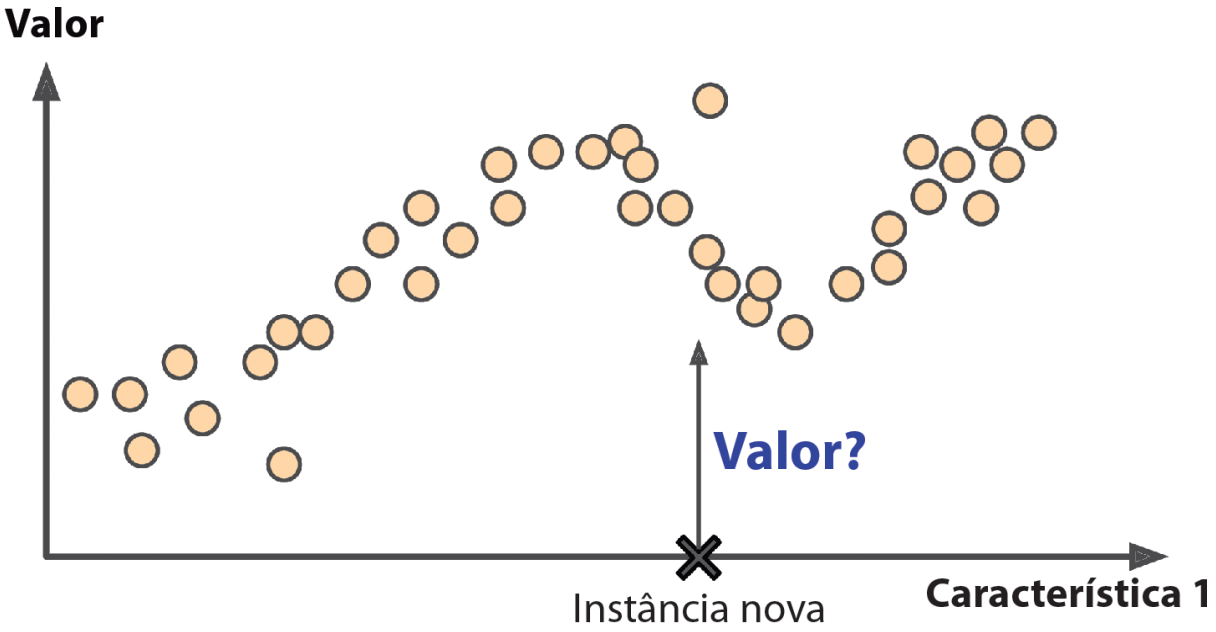
- No **aprendizado supervisionado**, o conjunto de treinamento que você fornece ao algoritmo inclui as soluções desejadas, chamadas de rótulos ou labels.

  Um conjunto de treinamento rotulado para **classificação** de spam (um exemplo de aprendizado supervisionado).



Aprendizado Supervisionado / Não Supervisionado

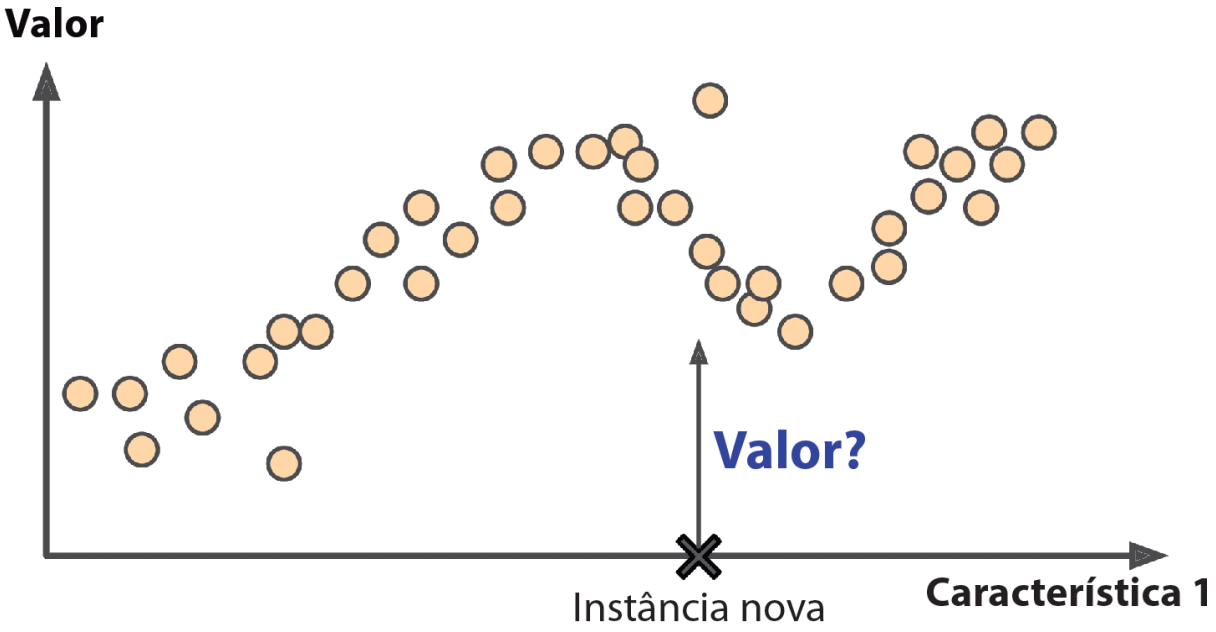
Um **problema de regressão**: prever um valor, dada a entrada de uma característica (geralmente existem diversas entradas de característica e, às vezes, diversos valores de saída).



Aprendizado Supervisionado / Não Supervisionado

Alguns algoritmos de regressão também podem ser utilizados para classificação e vice-versa.

- A **regressão logística** é comumente utilizada para classificação, pois consegue gerar um valor correspondente à probabilidade de pertencer a uma determinada classe.



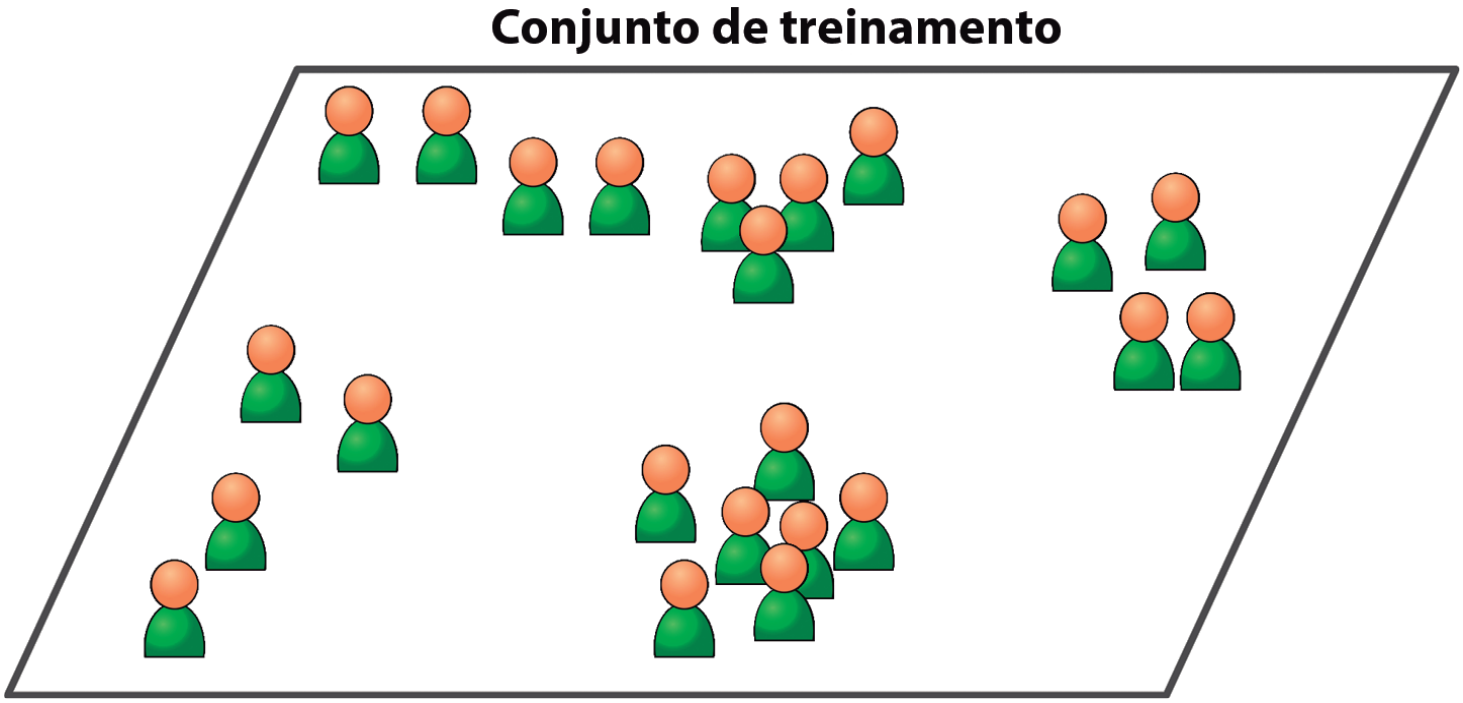
Aprendizado Supervisionado / Não Supervisionado

Algoritmos importantes de aprendizado supervisionado:

- K-ésimo vizinho mais próximo.
- Regressão linear.
- Regressão logística.
- Máquinas de vetores de suporte (SVMs).
- Árvores de decisão e florestas aleatórias.
- Redes neurais.

Aprendizado Não Supervisionado

Os dados de treinamento não são rotulados.



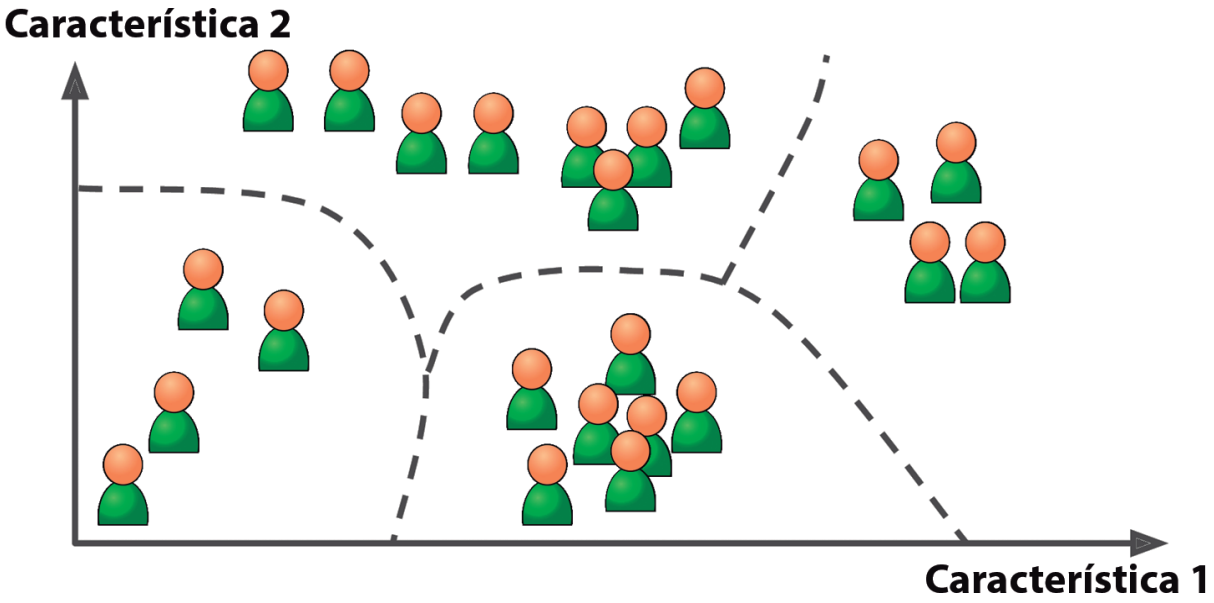
Aprendizado Não Supervisionado

Algoritmos importantes de aprendizado não supervisionado.

- **Clusterização**
  - K-Means (Clusterização K-média).
  - DBSCAN (clusterização espacial baseada em densidade de aplicações com ruído).
  - Análise de cluster hierárquica (HCA).
- **Detecção de anomalias e de novidades**
  - One-class SVM.
  - Floresta de isolamento.
- **Visualização e redução da dimensionalidade**
  - Análise de Componentes Principais (ACP).
  - Kernel ACP.
  - LLE (método de redução de dimensionalidade não linear [Locally Linear Embedding]).
  - t-SNE (método de incorporação estocástica de vizinhos distribuídos [Distributed Stochastic Neighbor Embedding]).
- **Aprendizado de regras por associação**
  - Apriori.
  - Eclat.

Aprendizado Não Supervisionado

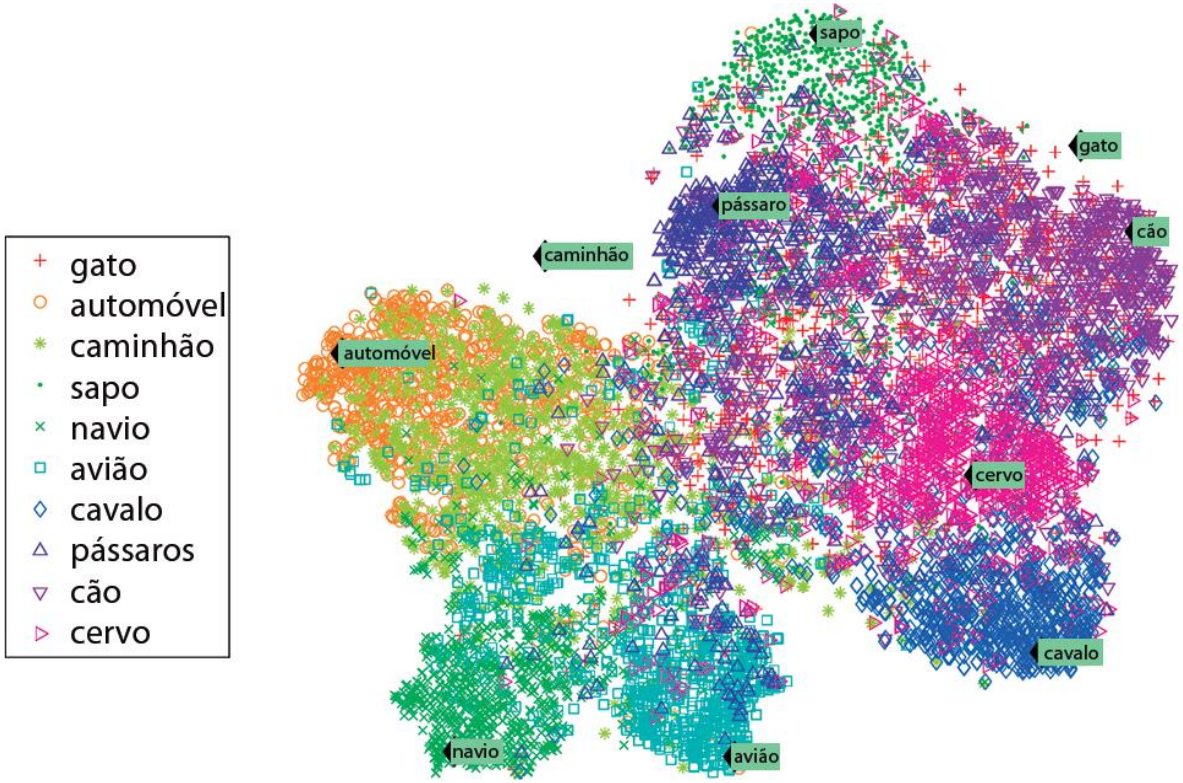
- **Clusterização**
  - Objetivo: tentar detectar grupos de visitantes semelhantes.
  - Em nenhum momento você informa ao algoritmo a qual grupo o visitante pertence.
    - Ele encontrará essas relações sem sua ajuda.



Aprendizado Não Supervisionado

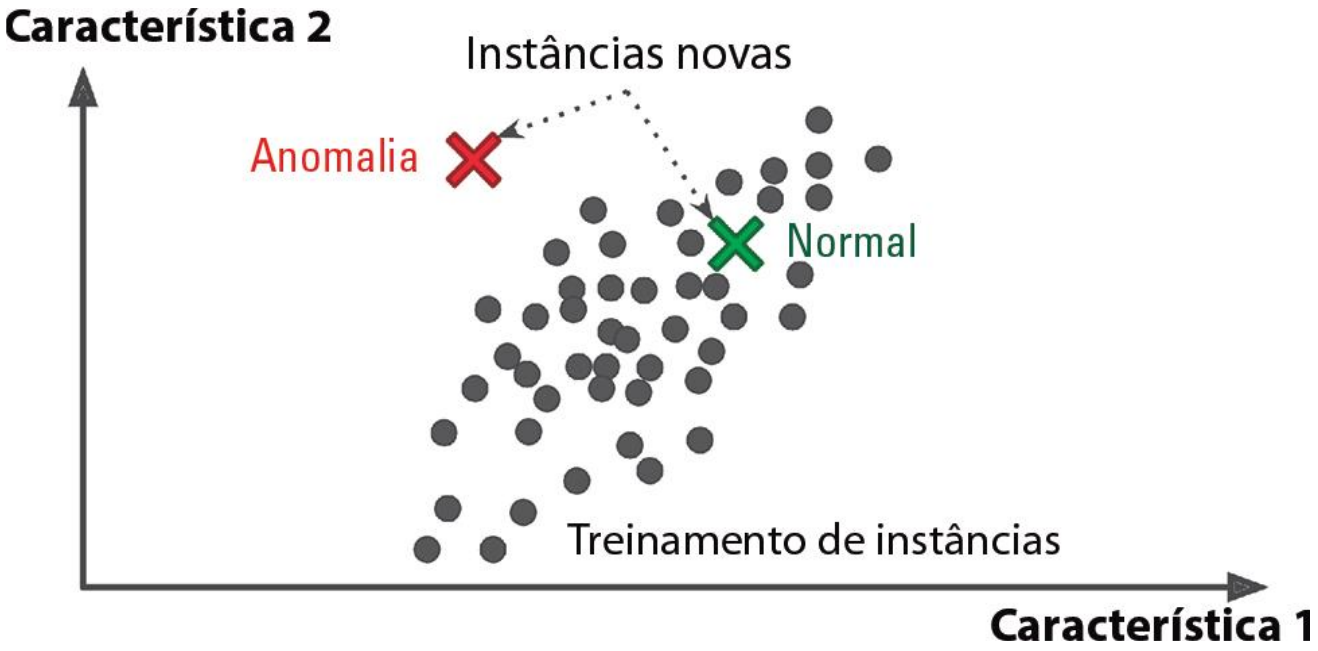
- **Algoritmos de visualização**

Visualização t-SNE destacando cluster semântico



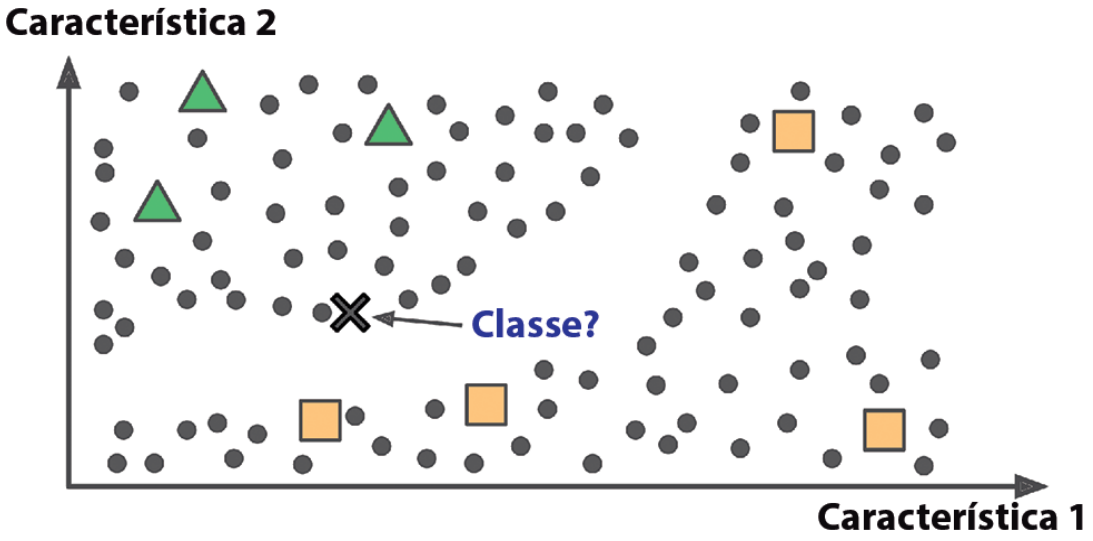
Aprendizado Não Supervisionado

- **Detecção de Anomalias**



Aprendizado semi-supervisionado

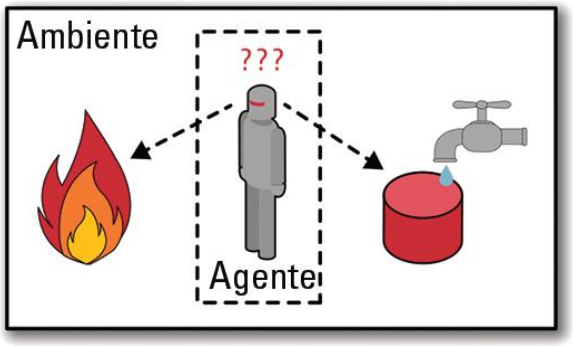
- Exemplos não rotulados (círculos) ajudam a classificar uma instância nova (a cruz) na classe triângulo em vez de na classe quadrado, ainda que esteja mais próxima dos quadrados rotulados.



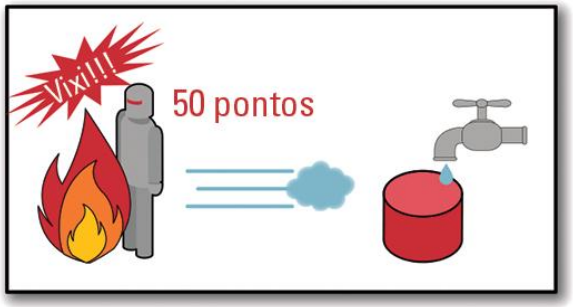
Aprendizado semi-supervisionado

- **Google Fotos**
  - **Clusterização** - Ao fazer o upload de todas as fotos de família, o aplicativo reconhecerá automaticamente que a mesma pessoa (A) aparece nas fotos 1, 5 e 11, enquanto outra pessoa (B) aparece nas fotos 2, 5 e 7.
  - **Classificação** - Acrescente somente um rótulo por pessoa e ele será capaz de nomear todas, o que é útil para pesquisar fotos.

Aprendizado por reforço



1. Assistir
2. Seleciona a ação usando a política



3. Mãos à obra!
4. Ganha uma recompensa ou penalidade

5. Atualização da política (etapa do aprendizado)
6. Faz iteração até encontrar uma política melhor

Aprendizado em batch e online

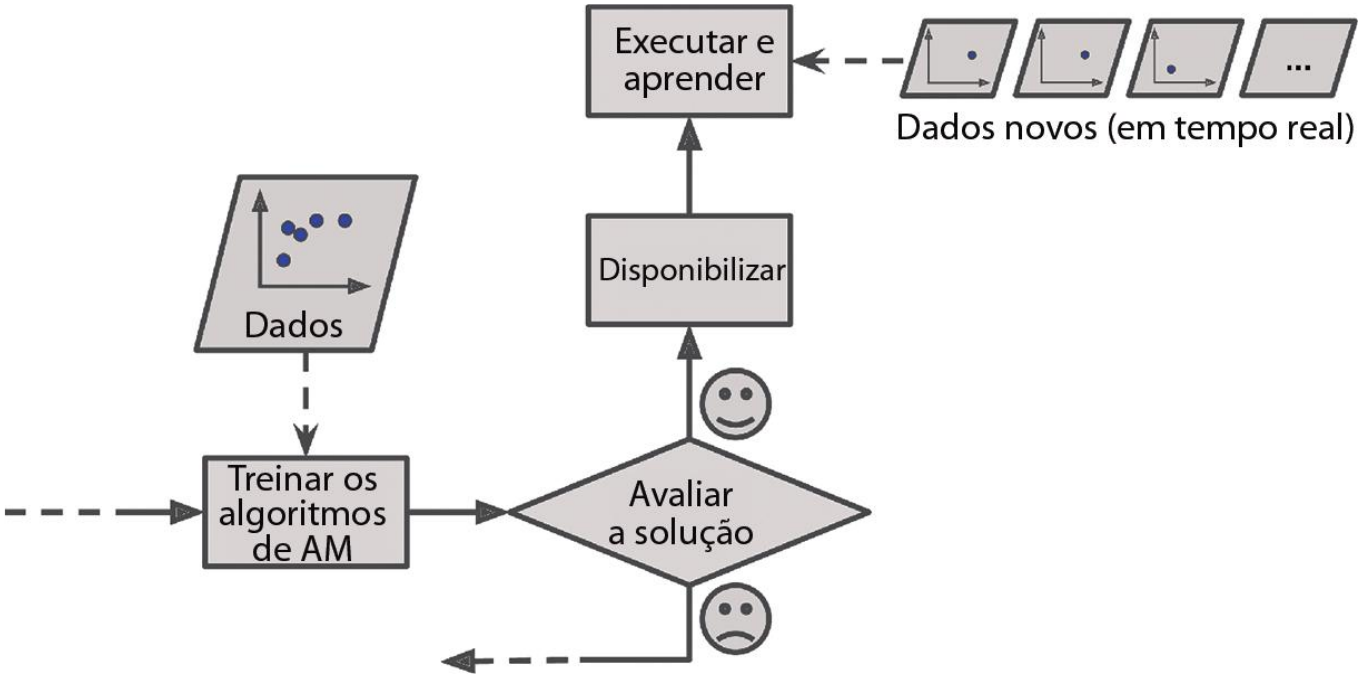
- Aprendizado em batch (por ciclo)
  - Modelo deve ser treinado usando todos os dados disponíveis.
  - O modelo é treinado e roda sem aprender mais nada, aplicando o que aprendeu (aprendizado offline).
  - Adaptação a mudanças: requer atualizar os dados e treinar uma nova versão do sistema a partir do zero sempre que necessário.

Aprendizado em batch e online

- Aprendizado online (incremental)
  - Permite treinar o sistema incrementalmente, fornecendo as instâncias de dados de forma sequencial, individual ou em pequenos grupos, chamados de mini-batches.
  - Cada etapa do aprendizado é rápida e tem um custo baixo, assim o sistema pode aprender instantaneamente os dados novos em tempo real, assim que eles entram.

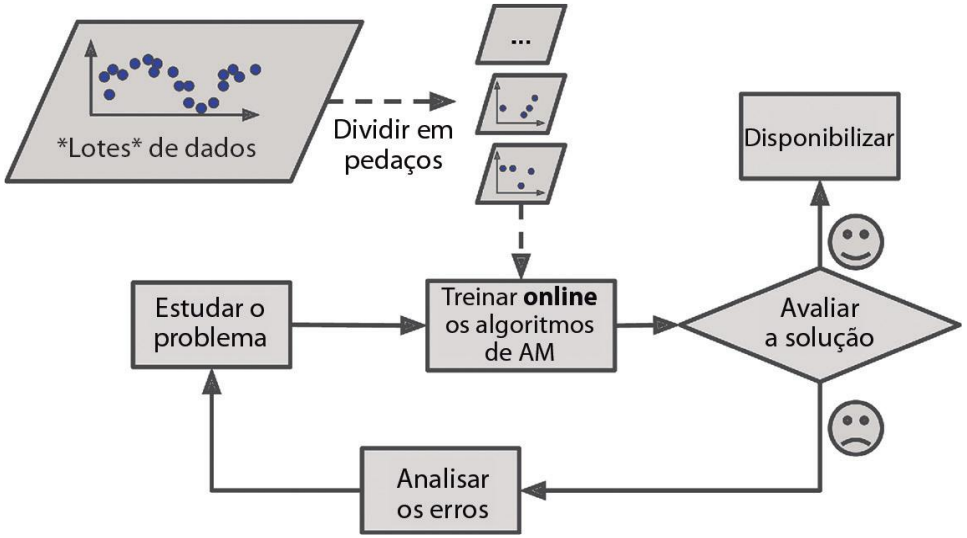
Aprendizado em batch e online

- Aprendizado online (incremental)



Aprendizado em batch e online

- Aprendizado online (incremental)
  - Aprendizado online também pode ser utilizado para treinar sistemas em grandes conjuntos de dados que não cabem na memória principal de uma máquina (aprendizado out-of-core).
  - O aprendizado out-of-core geralmente é feito offline. Melhor chamar: aprendizado incremental.



Aprendizado em batch e online

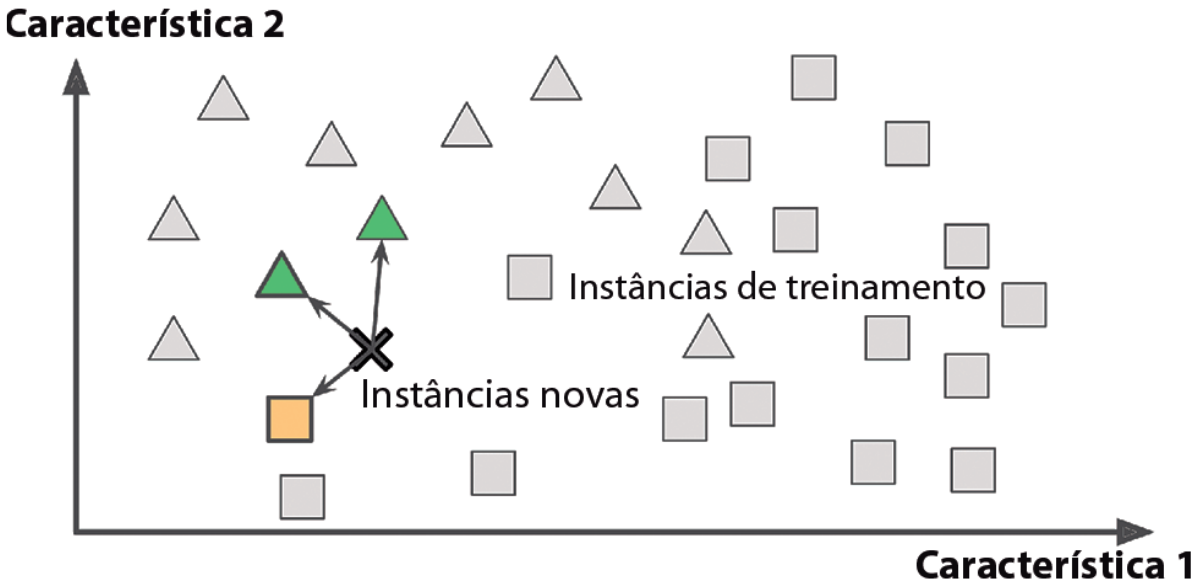
- Aprendizado online (incremental)
  - Parâmetro importante:
    - Taxa de aprendizagem
      - Rapidez com que eles devem se adaptar às mudanças dos dados.
      - Alta taxa de aprendizado: o sistema se adaptará rapidamente aos dados novos, mas também será propenso a se esquecer rapidamente dos antigos.

Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em instância
  - O modelo aprende os exemplos por meio da memorização e depois generaliza em novos casos, ao empregar uma medida de similaridade a fim de compará-los a outros exemplos aprendidos.
  - Exemplo:
    - O sistema marcaria um e-mail como spam se tivesse muitas palavras em comum com um e-mail de spam conhecido.

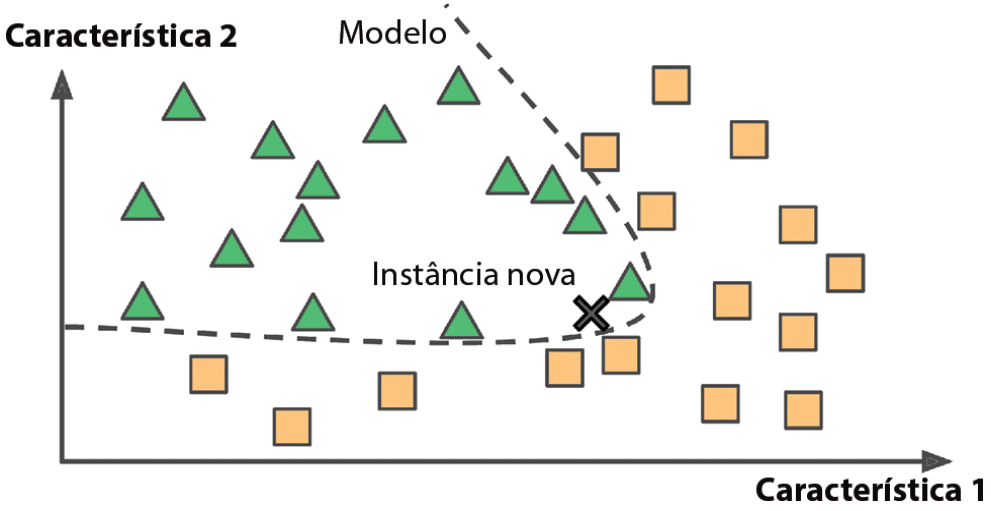
Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em instância



Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - generalização de um conjunto de exemplos para construir um modelo desses exemplos e usá-lo para fazer predições.



Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo

País	PIB por per capita (USD)	Satisfação de vida
Hungria	12.240	4,9
Coreia	27.195	5,8
França	37.675	6,5
Austrália	50.962	7,3
Estados Unidos	55.805	7,2

Pode-se modelar a satisfação de vida como uma função linear do PIB per capita:

satisfação_de_vida = $\theta_0 + \theta_1 \times$ PIB_per_capita

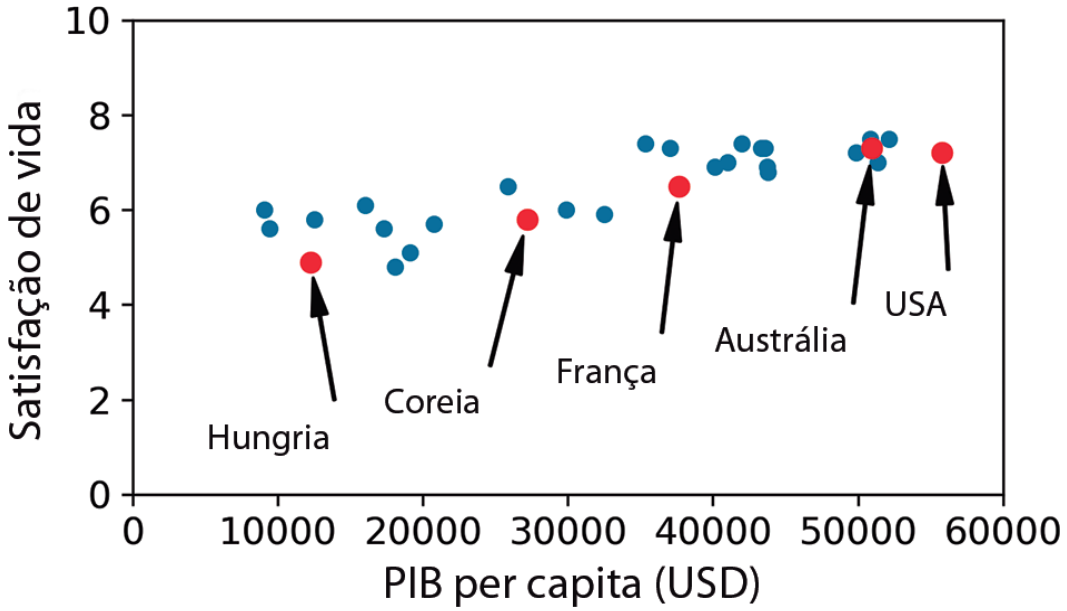
Antes de usar o modelo, é preciso definir os valores dos parâmetros θ_0 e θ_1.

Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - Para isso é necessário especificar uma medida de desempenho.
  - Você pode definir uma função de utilidade (ou função de avaliação) que calcula o quanto o seu modelo é bom, ou uma função de custo, que calcula o quanto ele é ruim.
  - Em problemas de regressão linear, geralmente utilizamos uma função de custo que calcula a distância entre as predições do modelo linear e os exemplos de treinamento.
    - Objetivo: minimizar essa distância.

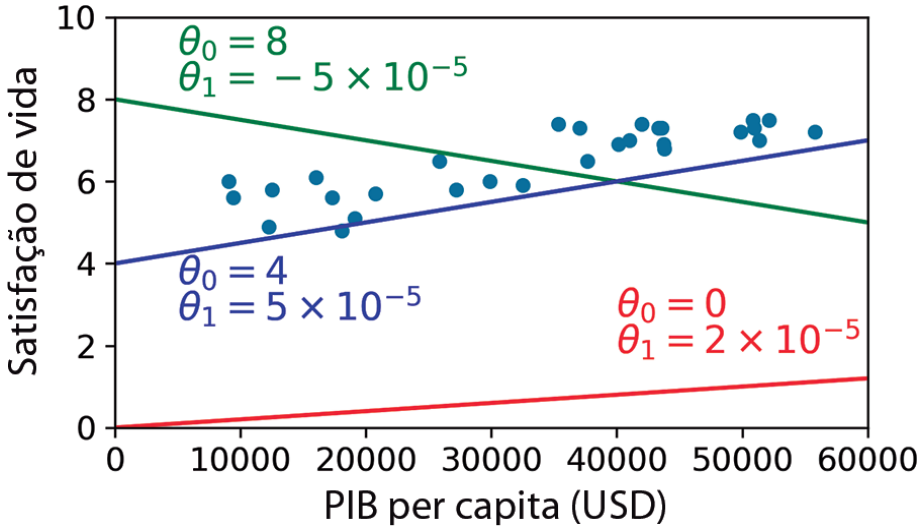
Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - Há uma tendência?



Aprendizado baseado em instâncias versus aprendizado baseado em modelo

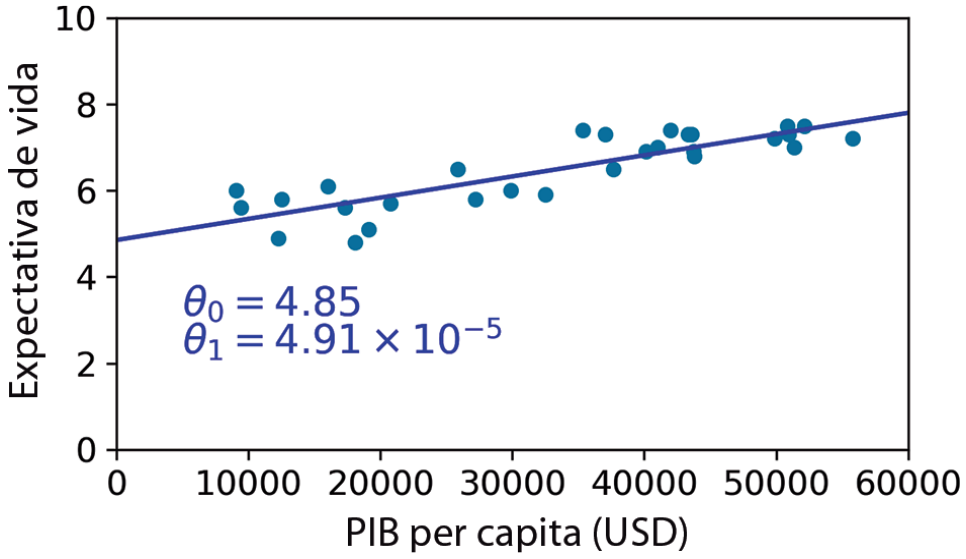
- Aprendizado baseado em modelo
  - Modelos lineares possíveis



Treinar um modelo significa executar um algoritmo a fim de identificar os parâmetros do modelo que melhor se adequem aos dados de treinamento (e, quem sabe, fazer boas predições a partir dos dados novos).

Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - Modelo linear que melhor se ajusta aos dados de treinamento



Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - https://github.com/ageron/handson-ml2/blob/master/01_the_machine_learning_landscape.ipynb

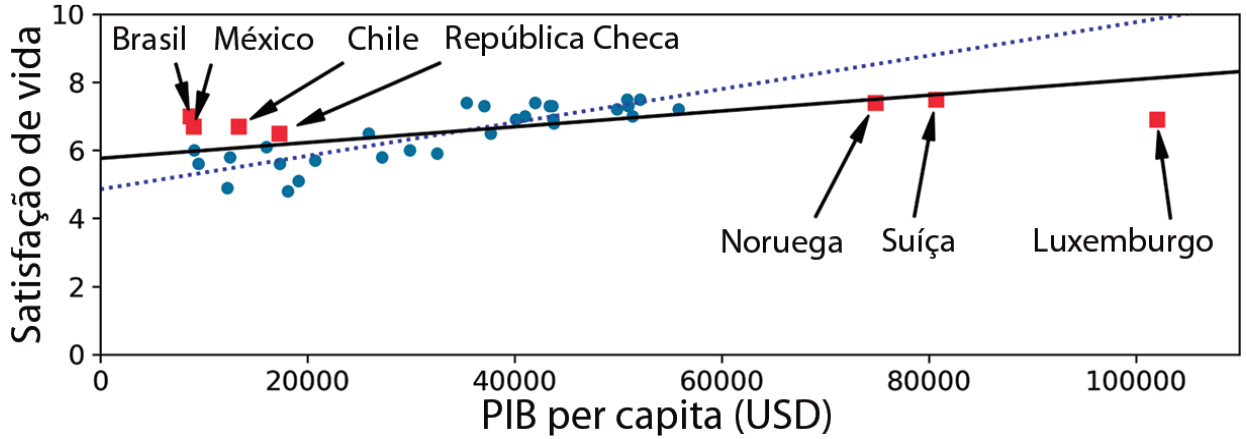
Seria possível substituir o modelo de regressão linear pelo algoritmo de regressão K-ésimo vizinho mais próximo.

Aprendizado baseado em instâncias versus aprendizado baseado em modelo

- Aprendizado baseado em modelo
  - Resumo:
    - Você estudou os dados.
    - Selecionou o modelo.
    - Treinou o modelo nos dados de treinamento (ou seja, o algoritmo de aprendizado procurou os valores dos parâmetros do modelo que minimizam uma função de custo).
    - E, por último, aplicou o modelo para fazer predições em novos casos (isso se chama inferência), na expectativa de que esse modelo fizesse boas generalizações.

Principais desafios do Aprendizado de Máquina

- Quantidade insuficiente de dados de treinamento.
- Dados de Treinamento Não Representativos.



Amostra de treinamento mais representativa.

Principais desafios do Aprendizado de Máquina

- Eficácia Irracional dos Dados
  - Devemos avaliar o custo-benefício entre gastar tempo e dinheiro no desenvolvimento de algoritmos ou empregá-los no desenvolvimento de corpus.
  - A ideia de que os dados são mais importantes do que os algoritmos em problemas complexos foi popularizada por Peter Norvig et al. em um artigo intitulado “The Unreasonable Effectiveness of Data” [“ A Eficácia Irracional dos Dados”] - 2009 (https:// homl.info/ 7).
  - No entanto, conjuntos de dados pequenos e médios ainda são muito comuns, e nem sempre é fácil ou barato obter dados extras de treinamento — portanto, não abra mão dos algoritmos ainda.

Principais desafios do Aprendizado de Máquina

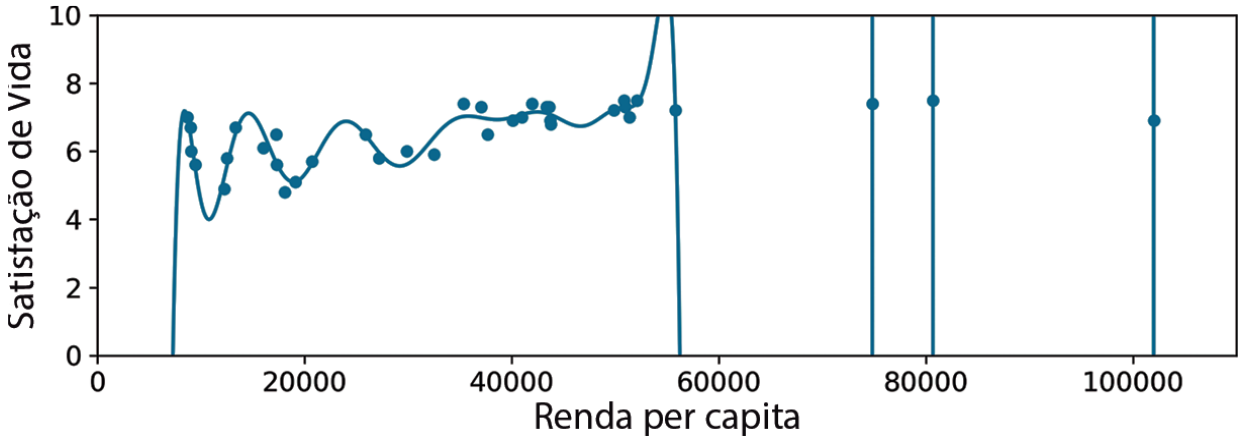
- Dados de baixa qualidade
  - Se seus dados de treinamento estiverem cheios de erros, outliers e ruídos o sistema terá mais dificuldade para detectar os padrões básicos.
  - Muitas vezes vale a pena dedicar um tempo limpando os dados de treinamento:
    - Se algumas instâncias são claramente outliers, apenas descartá-las pode ajudar, ou você pode tentar corrigir os erros manualmente.
    - Caso falte algumas características para algumas instâncias, você deve decidir:
      - se deseja ignorar completamente esse atributo
      - se deseja ignorar essas instâncias
      - preencher os valores ausentes (por exemplo, com a média da idade), ou
      - treinar um modelo com a característica e um modelo sem.

Principais desafios do Aprendizado de Máquina

- Características Irrelevantes
  - Entra lixo, sai lixo.
  - Seu sistema só será capaz de aprender se os dados de treinamento tiverem características relevantes suficientes e poucas características irrelevantes.
  - Criar um bom conjunto de características para o treinamento, processo chamado de feature engeneering (ou engenharia de features) que envolve os seguintes passos:
    - Seleção de características (selecionar as características mais úteis para treinamento entre as características existentes).
    - Extração de características (combinar características existentes a fim de obter as mais uteis — como vimos anteriormente, os algoritmos de redução de dimensionalidade podem ajudar).
    - Criação de novas características ao coletar dados novos.

Principais desafios do Aprendizado de Máquina

- Sobreajuste dos Dados de Treinamento
  - Generalizar as coisas exageradamente é algo que nós, humanos, fazemos com muita frequência, e infelizmente as máquinas podem cair na mesma armadilha se não tomarmos cuidado.
  - sobreajuste: o modelo funciona bem nos dados de treinamento, mas não generaliza tão bem.

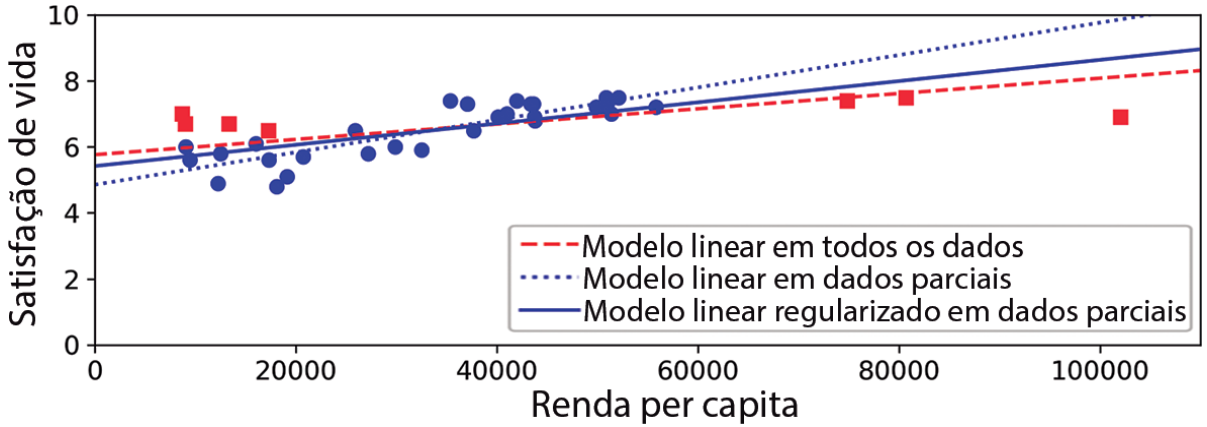


Principais desafios do Aprendizado de Máquina

- Sobreajuste dos Dados de Treinamento
  - Soluções:
    - Simplificar o modelo ao selecionar um com menos parâmetros (por exemplo, um modelo linear em vez de um modelo polinomial de alto nível), reduzindo o número de atributos nos dados de treinamento ou restringindo o modelo.
    - Coletar mais dados de treinamento.
    - Reduzir o ruído nos dados de treinamento (por exemplo, corrigir erros de dados e remover outliers).

Principais desafios do Aprendizado de Máquina

- Sobreajuste dos Dados de Treinamento
  - Chamamos de regularização quando restringimos um modelo para simplificar e reduzir o risco de sobreajuste.
    - A quantidade de regularização aplicada durante o aprendizado pode ser controlada por meio de um hiperparâmetro. Um hiperparâmetro é um parâmetro de um algoritmo de aprendizado (não do modelo).
    - Deve ser definido antes do treinamento e permanecer constante ao longo dele.



Principais desafios do Aprendizado de Máquina

- Subajuste dos dados de treinamento
  - ocorre quando o modelo é muito simples para o aprendizado da estrutura fundamental dos dados.
  - Exemplo: modelo linear de satisfação de vida está propenso a ser subajustado.
    - A realidade é mais complexa do que o modelo, por isso as predições tendem a ser imprecisas mesmo nos exemplos de treinamento.
  - Soluções:
    - Selecionar um modelo mais poderoso, com mais parâmetros.
    - Alimentar o algoritmo de aprendizado com melhores características (feature engineering).
    - Minimizar as restrições no modelo (por exemplo, reduzindo o hiperparâmetro de regularização).

Teste e Validação

- Treina-se o modelo utilizando o conjunto de treinamento e o testa usando o conjunto de teste.
- A taxa de erro nos casos novos se chama erro de generalização (ou erro fora da amostra).
- Ao avaliar o modelo no conjunto de teste, você obtém uma estimativa desse erro.
  - Esse valor lhe informa o desempenho do modelo em instâncias que ele nunca trabalhou antes.
- Sobreajuste: erro de treinamento baixo e erro de generalização alto.
- É comum usar 80% dos dados para treinamento e separar 20% para testes.

Ajuste de hiperparâmetro e seleção de modelo

- Problema: Calcular o erro de generalização diversas vezes sobre o conjunto de teste pode levar o modelo a não funcionar bem com os dados novos.
- Solução: Método *holdout* de validação cruzada
  - Dividir parte do conjunto de treinamento a fim de avaliar diversos modelos concorrentes e selecionar o melhor.
  - O novo conjunto separado se chama conjunto de validação (ou, às vezes, conjunto de desenvolvimento ou *dev set*).
  - Treinar modelos de dados com vários hiperparâmetros no conjunto de treinamento limitado (conj. treino completo – conj. validação) e seleciona o modelo com melhor desempenho no conj. validação.
  - Após esse processo holdout de validação cruzada, você treina o melhor modelo em todo o conjunto de treinamento (incluindo o conjunto de validação), e isso fornece o modelo final.
  - Por fim, você avalia esse modelo final no conjunto de testes para obter uma estimativa do erro de generalização.

Ajuste de hiperparâmetro e seleção de modelo

- Se conjunto de treino for pequeno, pode-se usar o método k-fold de validação cruzada:
  - Cada modelo é avaliado uma vez por conjunto de validação, após ser treinado no restante dos dados.
  - Ao calcular a média de todas as avaliações de um modelo, você obtém uma medida mais precisa de seu desempenho.
  - Inconveniente: o tempo de treinamento é multiplicado pelo número de conjuntos de validação.

Incompatibilide de dados

- Em algumas situações, é fácil obter uma quantidade massiva de dados para treinamento, mas esses dados não representarão perfeitamente os dados que serão usados na produção.
- O conjunto de validação e o conjunto de teste devem ser o mais representativos possível dos dados que você pretende usar em produção.

Teorema não existe almoço grátis

- Para alguns conjuntos de dados, o melhor modelo é um modelo linear, ao passo que, para outros conjuntos, será uma rede neural.
- Não existe um modelo que a priori funcione melhor.
- A única maneira de saber com certeza qual seria o melhor modelo é avaliar todos.
- Como isso é impossível, na prática você parte de alguns pressupostos sobre os dados e avalia somente alguns modelos razoáveis.
- Para tarefas simples, você pode avaliar modelos lineares com vários níveis de regularização e, para um problema complexo, pode avaliar diversas redes neurais.

Resumo

- **Aprendizado de máquina** é garantir que as máquinas evoluam em algumas tarefas aprendendo com os dados, em vez de ter que programar explicitamente as regras.
- Existem muitos tipos diferentes de sistemas AM:
  - **supervisionados ou não**
  - em **batch ou online**
  - **baseados em instâncias ou em modelos**
    - Baseados em modelos
      - ajusta alguns parâmetros para adequar o modelo ao conjunto de treinamento; se tudo der certo, também poderá fazer boas predições em novos casos.
    - Baseados em instâncias
      - Memoriza os exemplos e utiliza uma medida de similaridade para generalizar em instâncias novas.
- O modelo não deve ser simples demais (subajustado) nem muito complexo (superajustado).

Obrigado!

Dúvidas, comentários, sugestões?

Regis Pires Magalhães
regismagalhaes@ufc.br

UFC