ICP363
Introdução ao Aprendizado de Máquina

Aula 4 - Conjuntos de Treinamento e Teste, Reamostragem

Prof.a. Carolina G. Marcelino

Introdução

- Dada uma base de dados, existem diversos métodos de aprendizado de máquina que podemos utilizar para construir um modelo que possa prever/classificar novos dados.
- Em princípio, não há um método que possa ser considerado melhor que todos os outros, pois o desempenho de cada um deles pode variar de acordo com a base de dados que temos a nossa disposição.
- Por isso, é importante termos critérios de avaliação robustos para determinar qual modelo vamos utilizar.

Introdução

- Considere que temos:
  - uma variável de entrada (*feature*) X,
  - uma variável de saída Y,
  - uma base de dados formada por $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, onde $\{x_1, x_2 \cdots, x_n\}$ (resp., $\{y_1, y_2 \cdots, y_n\}$) são os valores que X (resp., Y) pode assumir.

  Regressão Linear: Modelo estatístico que examina a *relação linear* entre duas ou mais variáveis.

  Quando uma (ou mais) variável independente cresce (ou decresce), a variável dependente cresce (ou decresce).

- **Exemplos**
  - Peso x Pressão Sanguínea
  - Vendas x Anúncios
  - Horas de Estudo x Nota

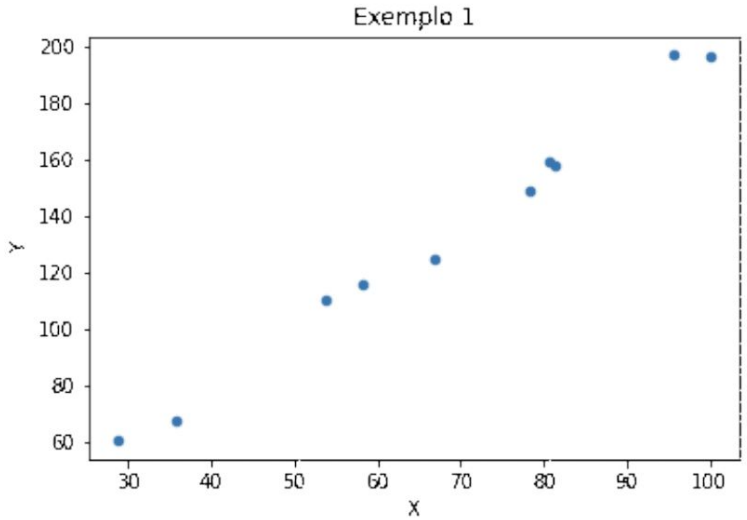
Regressão Linear

- **Objetivo**: determinar a função $\hat{f}$, que dado o valor de entrada x' preveja o valor de saída $y' = \hat{f}(x')$.

X	Y
95.724162408	197.179636092
35.7576189281	67.5906695414
28.8168474238	60.8541328206
99.9584813087	196.907396981
66.8097483121	125.311128524
58.2156926413	115.785784589
53.8210763379	110.762772705
81.2960821704	157.98528569
80.6486970595	159.61941373
78.2528136925	149.003865539

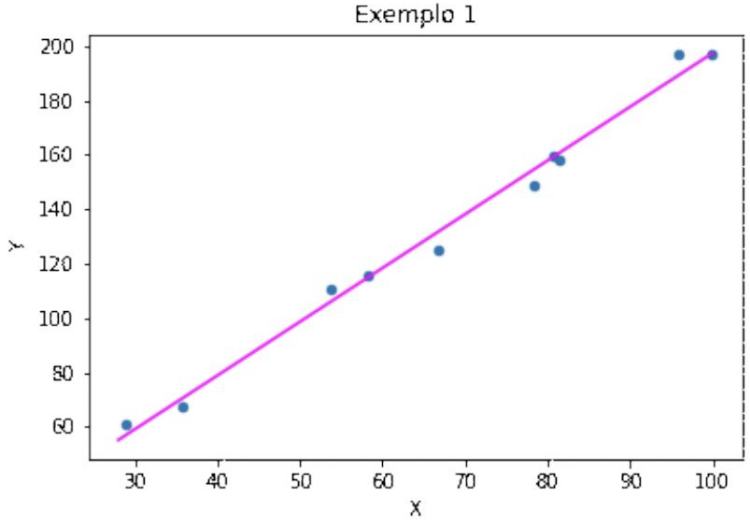
Regressão Linear

- **Objetivo**: determinar a função $\hat{f}$, que dado o valor de entrada x' preveja o valor de saída $y' = \hat{f}(x')$.



Regressão Linear

- **Objetivo**: determinar a função $\hat{f}$, que dado o valor de entrada x' preveja o valor de saída $y' = \hat{f}(x')$.



Escolha do Modelo

Para avaliar o desempenho de um método de aprendizado estatístico em um dado conjunto de dados, precisamos de alguma forma de medir o quão bem suas previsões realmente correspondem aos dados observados.

Ou seja, precisamos quantificar até que ponto o valor de resposta previsto para uma dada observação está próximo do valor de resposta verdadeiro para essa observação. Na configuração de regressão, a medida mais comumente usada é o erro quadrático médio (MSE), dado por:

- **Medida**: *Erro Quadrático Médio (Mean Squared Error - MSE)*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

Escolha do Modelo

Na configuração de regressão, a medida mais comumente usada é o erro quadrático médio (MSE), dado por:

- **Medida**: *Erro Quadrático Médio (Mean Squared Error - MSE)*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

onde **^f(xi)** é a previsão que **^f** dá para a **i-ésima** observação.

O MSE será pequeno se as respostas previstas forem muito próximas das respostas verdadeiras,

O MSE será grande se para algumas das observações, as respostas previstas e verdadeiras diferirem substancialmente.

Escolha do Modelo

Objetivo: determinar a função $\hat{f}$, dado o valor de entrada x' preveja o valor de saída $y' = \hat{f}(x')$.

- **Medida**: *Erro Quadrático Médio (Mean Squared Error - MSE)*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

  Quanto menor o valor de MSE, melhor a previsão que está sendo feita.

- Observe que este erro está sendo medido com relação aos valores (x_i, y_i) que **conhecemos**.

- Dizemos que o conjunto de dados (**conhecidos**) $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ é o nosso **conjunto de treinamento**.

- E vamos representar seu erro quadrático médio como MSE_{train}.

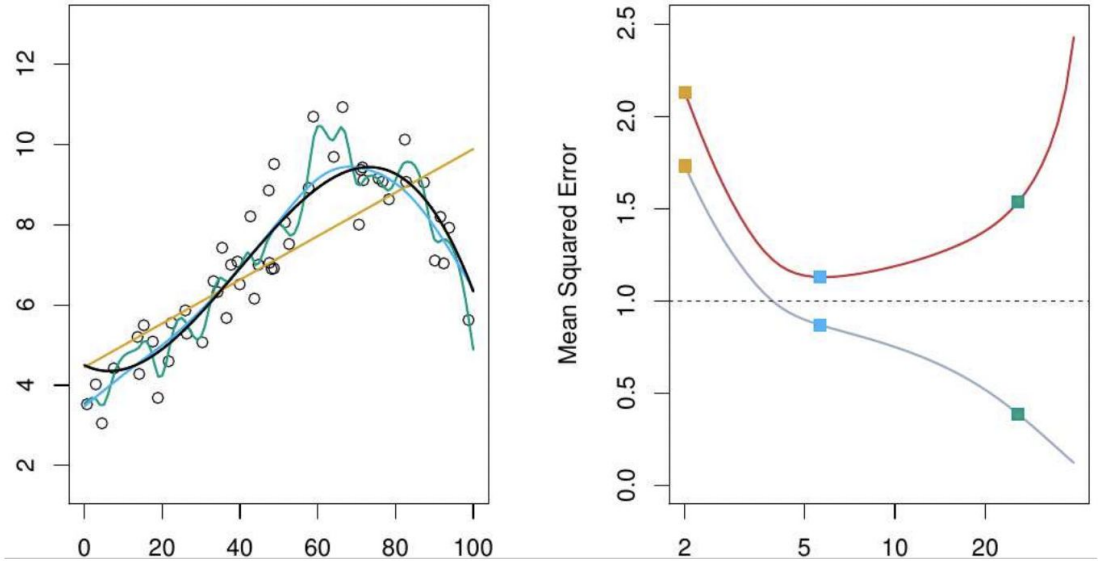
Escolha do Modelo

- Mas, em geral, não nos importamos realmente com o quão bem o método funciona treinando o MSE nos dados de treinamento.
- Em vez disso, estamos interessados na precisão das previsões que obtemos quando aplicamos nosso método a dados de teste nunca vistos antes.
- Por que é isso que nos importa?

- Suponha que estejamos interessados em dados de teste no desenvolvimento de um algoritmo para prever o preço de uma ação com base em retornos de ações anteriores. Podemos treinar o método usando retornos de ações dos últimos 6 meses. Mas não nos importamos realmente com o quão bem nosso método prevê o preço das ações da semana passada. Em vez disso, nos importamos com o quão bem ele preverá o preço de amanhã ou o preço do mês que vem.

Escolha do Modelo

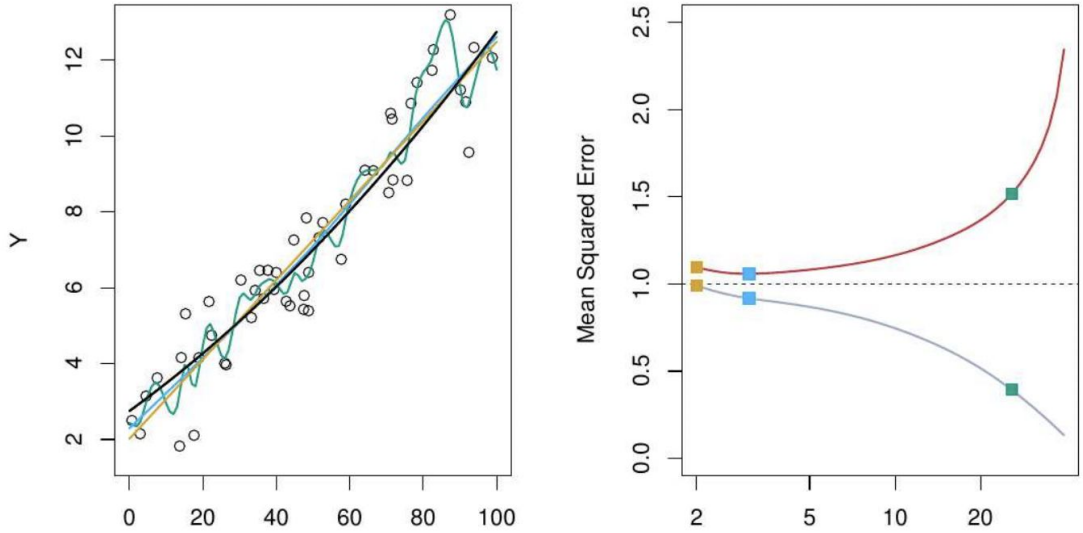
Caso contrário, escolhemos o modelo que possua o menor valor de MSE_{train}? MSE_{train} pequeno não implica em MSE_{test} pequeno



- Esquerda: Dados simulados de f, mostrados em preto. Três estimativas de f são mostradas: a linha de regressão linear (curva laranja) e duas splines de suavização (curvas azul e verde). Direita: MSE de treinamento (curva cinza), MSE de teste (curva vermelha) e MSE de teste mínimo possível sobre todos os métodos (linha tracejada). Os quadrados representam os MSEs de treinamento e teste para os três mostrados no painel esquerdo.

Escolha do Modelo

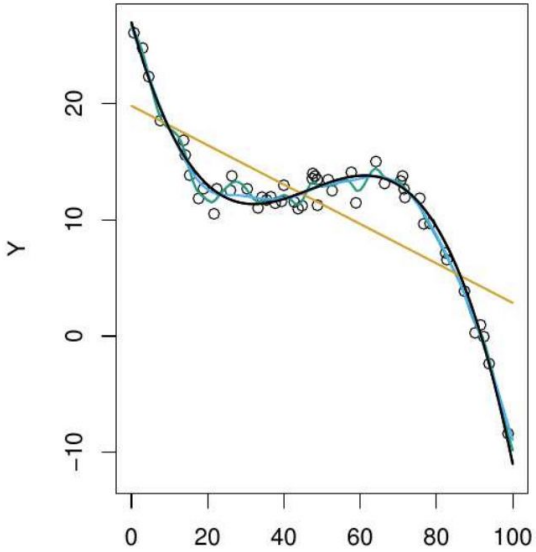
No gráfico à direita, MSE_{train} cai à medida que o modelo vai se adaptando aos dados, enquanto MSE_{test} começa a piorar em um dado momento - **overfitting**.

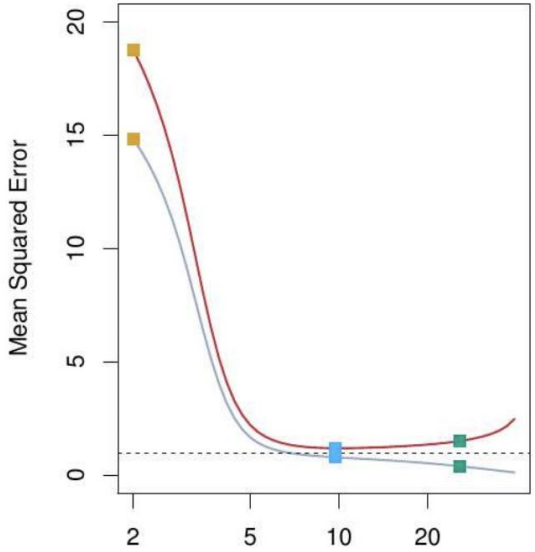


- **Overfitting** significa criar um modelo que corresponda (memorize) ao conjunto de treinamento de forma tão próxima que o modelo não consegue fazer previsões corretas em novos dados

Escolha do Modelo

No gráfico à direita, MSE_{train} cai à medida que o modelo vai se adaptando aos dados, enquanto MSE_{test} começa a piorar em um dado momento - **overfitting**.





- **Overfitting** significa criar um modelo que corresponda (memorize) ao conjunto de treinamento de forma tão próxima que o modelo não consegue fazer previsões corretas em novos dados

Escolha do Modelo

- Para declarar de forma mais matemática, suponha que ajustamos nosso método de aprendizado estatístico em nossas observações de treinamento **{(x1, y1),(x2, y2), . . . ,(xn, yn)}**, e obtemos a estimativa **ˆf**.

- Podemos então calcular **ˆf(x1), ˆf(x2), . . . , ˆf(xn)**.

- Se estes forem aproximadamente iguais a **y1, y2, . . . , yn**, então o MSE de treinamento é pequeno.

Escolha do Modelo

- No entanto, não estamos realmente interessados em saber se **ˆf(xi) ≈ yi**;

- em vez disso, queremos saber se **ˆf(x0)** é aproximadamente igual a **y0**, onde **(x0, y0)** é uma observação de teste não vista anteriormente não usada para treinar o método de aprendizado estatístico

Escolha do Modelo

- Queremos escolher o método que fornece o **menor MSE de teste**, em oposição ao menor **MSE de treinamento**. Em outras palavras,

$$\text{Ave}(y_0 - \hat{f}(x_0))^2,$$

- o erro de predição quadrático médio para essas observações de teste **(x0, y0)**. Gostaríamos de selecionar o modelo para o qual essa quantidade é a menor possível.

Escolha do Modelo

- Como podemos tentar selecionar um método que minimize o teste MSE de teste?
- Em algumas configurações, podemos ter um conjunto de dados de teste disponível — isto é, podemos ter acesso a um conjunto de observações que não foram usadas para treinar o método de aprendizado estatístico.
- Podemos então simplesmente avaliar **Ave (y0 - ^f(xo))²** nas observações de teste e selecionar o método de aprendizado para o qual o teste MSE é menor.

$$\text{Ave}(y_0 - \hat{f}(x_0))^2,$$

Escolha do Modelo

- É possível mostrar que o MSE de teste esperado, para um dado valor **x0**, pode sempre ser decomposto na soma de três quantidades fundamentais: a **variância de ˆf(x0)**, o **viés quadrado de ˆf(x0)** e a **variância do erro** termos **ε**.

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

- Aqui, a notação **E (y0 − ˆf(x0))²** define o MSE de teste esperado em **x0**, e se refere ao **MSE de teste médio** que obteríamos se testássemos repetidamente o **MSE** estimado **f** usando um grande número de conjuntos de treinamento e testássemos cada um em **x0**.

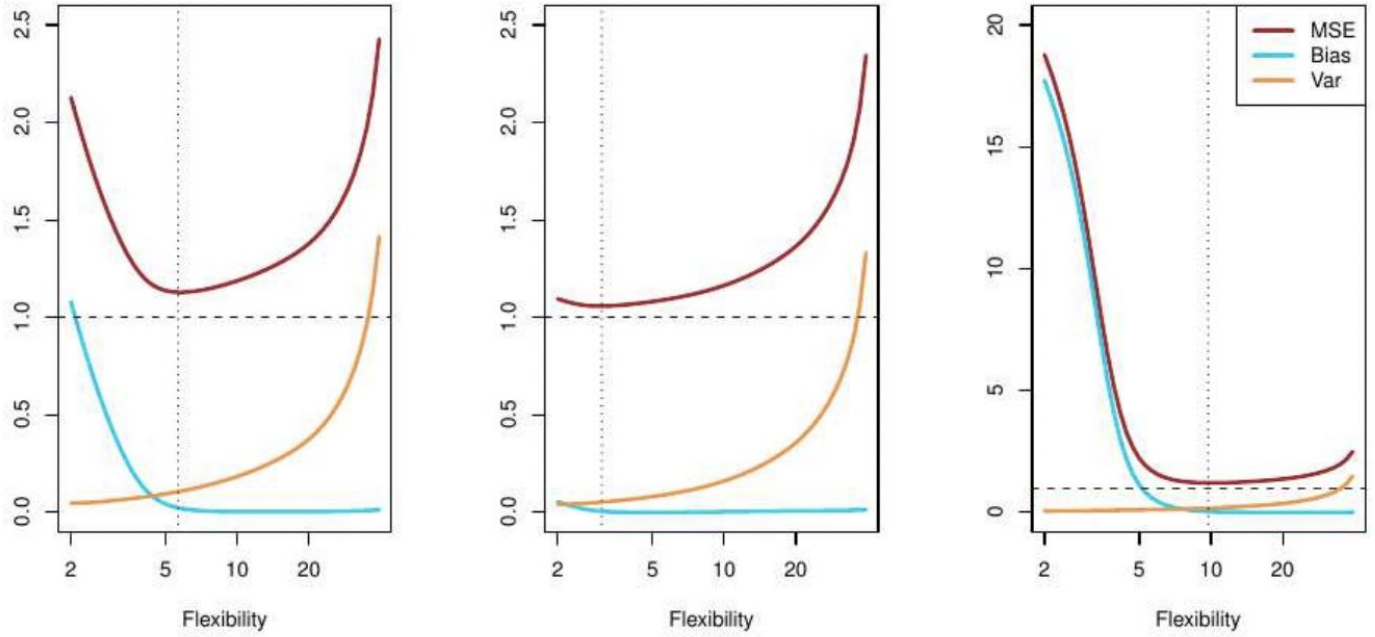
The Bias-Variance Trade-off

- Para um dado valor x_0, o valor esperado de MSE_{test} é composto por:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + (Bias(\hat{f}(x_0)))^2 + Var(\epsilon)$$

  - **Variância de $\hat{f}(x_0)$:** quantidade pela qual $\hat{f}$ mudaria se utilizássemos um conjunto de treinamento diferente. Se ela for alta, pequenas mudanças no conjunto de treinamento implicam grandes mudanças em $\hat{f}$. Quanto mais *flexível* o método, maior a variância.
  - **Viés ao quadrado de $\hat{f}(x_0)$**: erro introduzido ao modelarmos um problema muito complexo usando um modelo simples. Neste caso, métodos mais *flexíveis*, resultam em um viés menor.
  - **Variância do erro ϵ**: erro irredutível, uma vez que por mais perfeito que seja nosso modelo, ele sempre terá algum erro embutido (vindo de variáveis desconhecidas).
- Métodos mais flexíveis $\Rightarrow$ variância cresce e viés decresce.
- **Bias-Variance Trade-Off**: à medida que variância e viés mudam, impactam o crescimento ou a diminuição do MSE_{test}. O desafio é encontrar um método em que ambas (variância e viés) sejam baixos.

The Bias-Variance Trade-off



The Bias-Variance Trade-off

- Para um bom desempenho do modelo em um conjunto de testes, o método de aprendizado deve ter baixa variância e baixo viés.
- Fácil obter baixo viés e alta variância ou alto viés e baixa variância.
- O desafio é encontrar um método que tenha baixo viés e variância.
- Isso não é possível uma vez que normalmente f é desconhecido.

Classificação

- Considere que temos:
  - uma variável de entrada (*feature*) X,
  - uma variável de saída Y *qualitativa*,
  - uma base de dados formada por $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, onde $\{x_1, x_2 \cdots, x_n\}$ (resp., $\{y_1, y_2 \cdots, y_n\}$) são os valores que X (resp., Y) pode assumir.
- Podemos estimar o **erro** do nosso **classificador** determinando a proporção de classificações erradas que são feitas no conjunto de treinamento:

$$\frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

com

$$I(y_i \neq \hat{y}_i) = \begin{cases} 1 & \text{se } y_i \neq \hat{y}_i \\ 0 & \text{se } y_i = \hat{y}_i \end{cases}$$

Classificação - K vizinhos mais próximos

Ideia:

Memorizar o conjunto de treinamento e depois predizer o rótulo de qualquer nova instância com base nos rótulos de seus vizinhos mais próximos no conjunto de treinamento

- Algoritmo Supervisionado: o conjunto de treinamento precisa apresentar rótulos.
- Não paramétrico: não exige que os dados sigam uma distribuição

Classificação - K vizinhos mais próximos

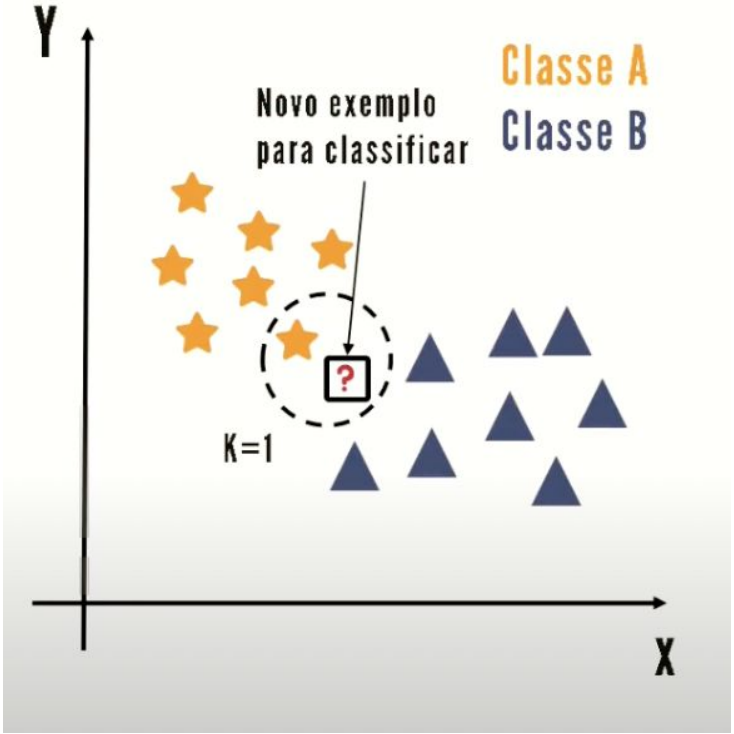
Ideia:

Memorizar o conjunto de treinamento e depois predizer o rótulo de qualquer nova instância com base nos rótulos de seus vizinhos mais próximos no conjunto de treinamento

- Preguiçoso: porque não há uma etapa de treinamento propriamente dita. Cada nova observação é comparada (via cálculo da distância) a cada uma das instâncias do conjunto de treinamento. Baseado nas observações. Calcula-se essa distância para cada observação.

Classificação - 1 vizinho mais próximo 1NN

- Calcula as distâncias entre cada dois pontos
- Um ponto é rotulado do treinamento e outro é o que desejamos rotular
- O ponto a ser rotulado recebe o rótulo do exemplo de treinamento mais próximo



Classificação - KNN

O algoritmo pressupõe que os atributos são numéricos

- Qualitativos: converter
- Quantitativos com escalas diferentes: normalizar
- Medidas de distância são afetadas pela escala dos atributos

Idade	Peso	Vacina	**Doença**
25	80	S	N
40	85	N	S
29	70	S	N
45	75	N	S

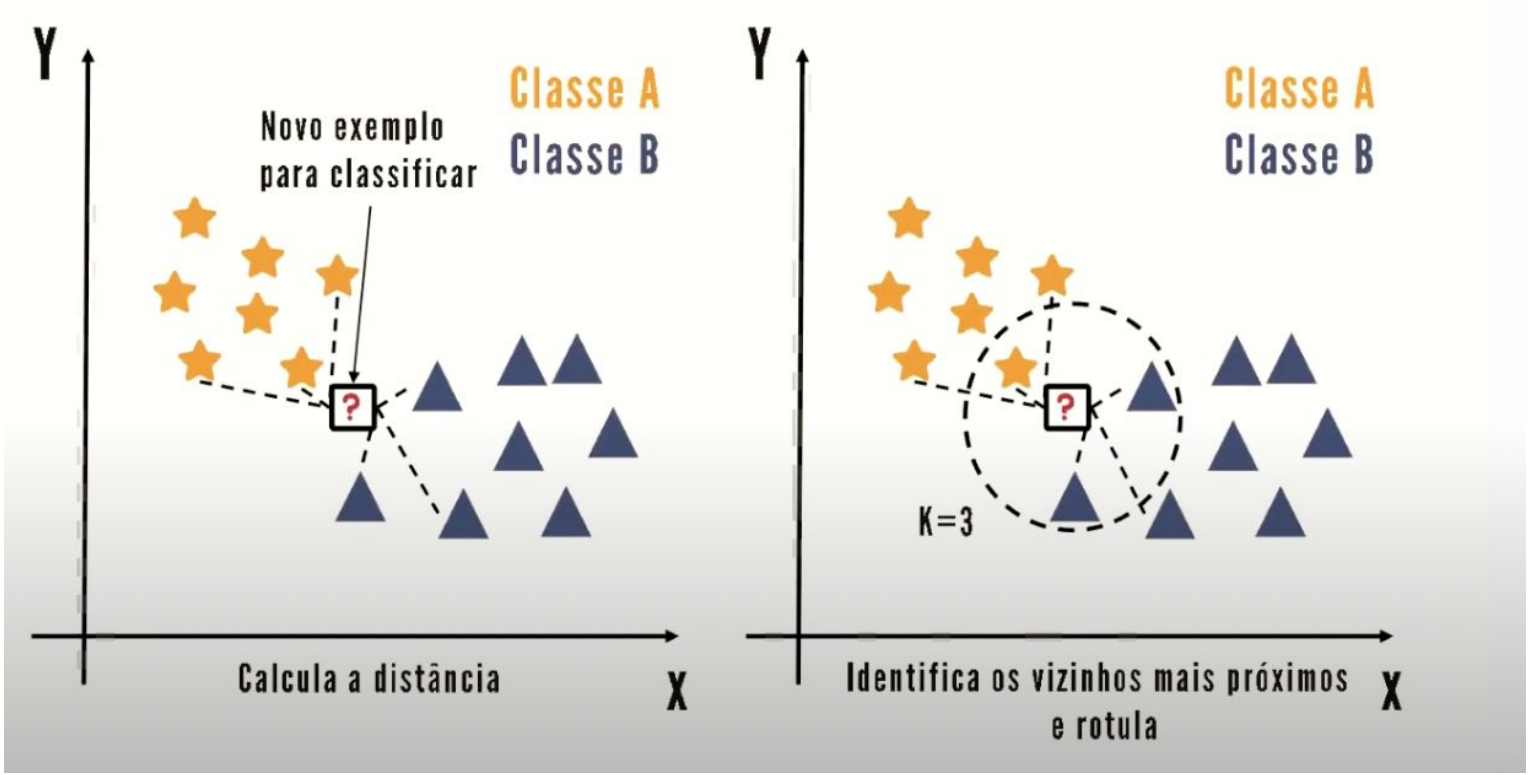
Classificação -Utilização KNN

- Conjunto de exemplos de treinamento
- Definir uma métrica para calcular a distância entre os exemplos de treinamento
- Definir o valor de K (é número de vizinhos mais próximos)

Classificação -Utilização KNN

- Conjunto de exemplos de treinamento
- Definir uma métrica para calcular a distância entre os exemplos de treinamento
- Definir o valor de K (é número de vizinhos mais próximos)

Classificação -Utilização KNN



Classificação -Preparando dados para o KNN

Normalização de dados

- Funciona melhor se todos os dados estiverem na mesma escala
- Normalizar os dados para o intervalo [0,1] é uma boa ideia

Lidar com dados ausentes

- Dados ausentes significam que a distância entre as amostras não pode ser calculada
- Essas amostras podem ser excluídas ou os valores ausentes podem ser imputados

Classificação - Medidas de Distância

Objetivo de calcular distâncias

- Determinar quais das K instâncias do conjunto de dados de treinamento são mais semelhantes a uma nova entrada
- Para variáveis de entrada de valor real, a medida de distância mais popular é a **distância euclidiana**

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^{d} (x_i^l - x_j^l)^2}$$

x_i, x_j: dois objetos representados por vetores do espaço $\Re^d$;
x_i^l, x_j^l: são elementos desses vetores, que correspondem aos valores da coordenada l (atributos).

Classificação - Como escolher o valor de K

Número par de Classes: K igual a um número ímpar

- Empates utilizar a classe da instância mais próxima

K Muito Grande

- Vizinhos podem ser muito diferentes
- Predição tendenciosa para a classe majoritária

K Muito pequeno

- Apenas objetos muito parecidos são considerados
- Predição pode ser instável

Classificação - Como escolher o valor de K

Número par de Classes: K igual a um número ímpar

- Empates utilizar a classe da instância mais próxima

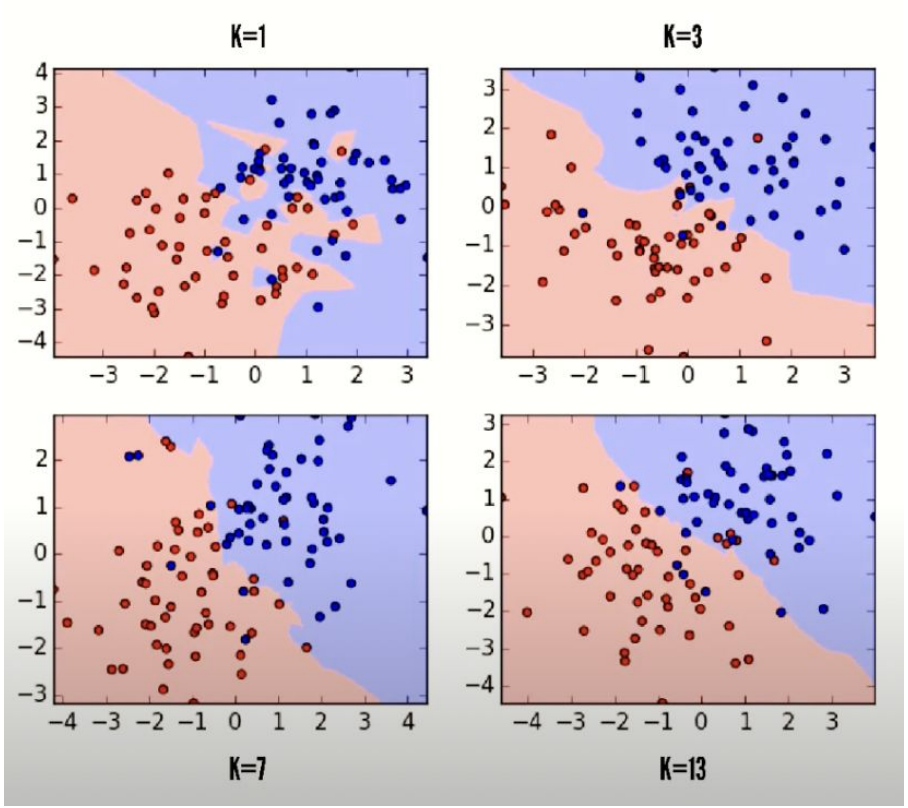
K Muito Grande

- Vizinhos podem ser muito diferentes
- Predição tendenciosa para a classe majoritária

K Muito pequeno

- Apenas objetos muito parecidos são considerados
- Predição pode ser instável

Classificação - Como escolher o valor de K



Classificação - Conjuntos de Treinamento e Teste

- Não devemos usar toda a nossa base de dados para treinar nosso modelo.
- Precisamos que alguns dados não sejam usados na fase de construção do modelo, para que possamos ajustar os parâmetros do modelo e avaliar o seu desempenho para dados que ele não conhece. Assim, tentamos evitar que nosso modelo fique enviesado.
- Nosso conjunto de dados deve ser dividido em um conjunto usado para treinar nosso modelo (*conjunto de treinamento*) e outro usado para avaliar a performance do modelo (*conjunto de teste ou validação*).
- Implicitamente estamos assumindo que temos um conjunto de dados grande o suficiente para ser considerado como representativo do domínio do nosso problema.
- Se nosso dataset for pequeno (tiver poucas amostras), teremos um conjunto de treinamento pequeno, o que pode impedir a construção do nosso modelo.

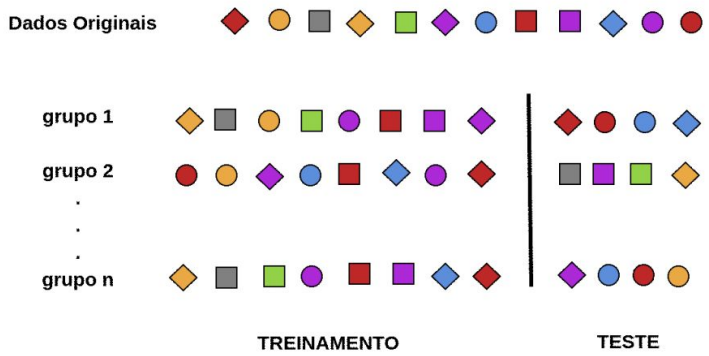
Classificação - Conjuntos de Treinamento e Teste

- Nosso conjunto de dados pode ser dividido em $\frac{2}{3}$ para treino e $\frac{1}{3}$ para teste ou 80% para treino e 20% para teste.
- A divisão dos conjuntos deve ser feita de forma aleatória.
- No caso de problemas de classificação, nosso conjunto de dados pode estar desbalanceado com relação as classes (ou seja, podemos ter mais exemplos de uma classe do que de outra). É desejável que ao dividir nosso conjunto de dados em treinamento e teste, a mesma proporção de exemplos por classe seja preservada em ambos os conjuntos.
- Um problema que esta abordagem apresenta é que usar um único conjunto de teste pode ser insuficiente para se ter uma boa avaliação.
- Para contornar este problema utiliza-se métodos de reamostragem que em geral produzem estimativas de performance melhor do que usar um único conjunto de teste, uma vez que são avaliados diversas versões alternativas dos dados.

Classificação - Conjuntos de Treinamento e Teste

Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **Repetição de Divisões em Treinamento e Teste**: esta abordagem simplesmente repete a criação de conjuntos de treinamento e teste várias vezes.



- O número de repetições é importante. Quanto mais amostras o conjunto de teste tiver, maior deverá ser o número de repetições, de modo a diminuir a incerteza da estimativa de performance. Caso o conjunto de dados seja pequeno, a variância de desempenho pode ser grande.

Classificação - Conjuntos de Treinamento e Teste

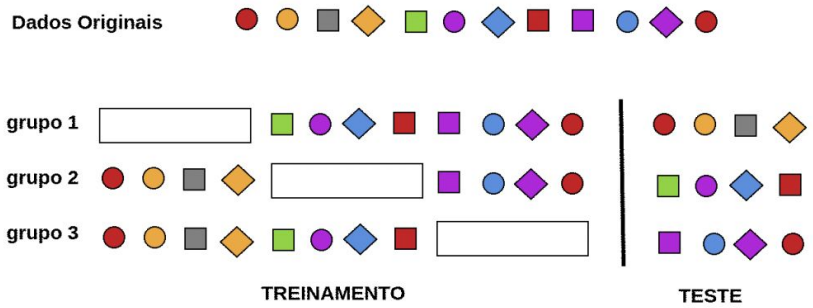
Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **K-Fold Cross Validation**: o conjunto de amostras é dividido aleatoriamente em K subconjuntos de mesmo tamanho.

  O modelo é treinado usando todos os subconjuntos exceto o primeiro, que é usado na avaliação do modelo gerado.

  O processo é repetido, sendo que agora o segundo subconjunto é usado para avaliar o modelo, e os demais no treinamento.

  A performance do modelo é medida pela média de performance do K modelos.



Classificação - Conjuntos de Treinamento e Teste

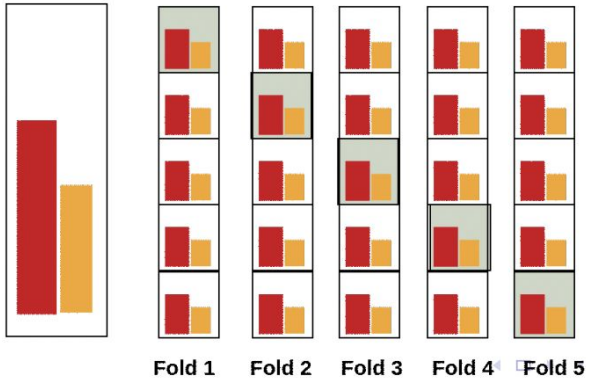
Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **K-Fold Cross Validation**
  - Valores comuns para K são 3, 5 e 10.
  - Avaliar o mesmo modelo com diferentes valores de K e compará-los.
  - O ideal seria treinar o modelo com todo o dataset e avaliar o desempenho em um conjunto que não foi usado no treinamento. Mas isso normalmente não é viável.
  - Uma alternativa é usar o leave-one-out cross-validation (LOOCV), que significa fazer $K = n$, onde n é o número de instâncias no dataset. Computacionalmente é muito custosa.
  - Pode-se comparar a precisão média da classificação para diferentes valores de K com a precisão média da classificação de LOOCV no mesmo conjunto de dados.
  - A diferença entre as pontuações fornece uma aproximação de quão bem um valor K se aproxima da condição de teste de avaliação do modelo ideal.

Classificação - Conjuntos de Treinamento e Teste

Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **Stratified K-Fold Cross Validation**
  - Usar K-fold Cross Validation no caso do conjunto de dados ser desbalanceado, o modelo pode ser impactado. O Stratified K-Fold Cross Validation procura atacar este problema.
  - No caso de problemas de classificação, a divisão de dados é feita de forma que cada subconjunto tenha a mesma proporção de elementos de cada classe que tem no conjunto todo.

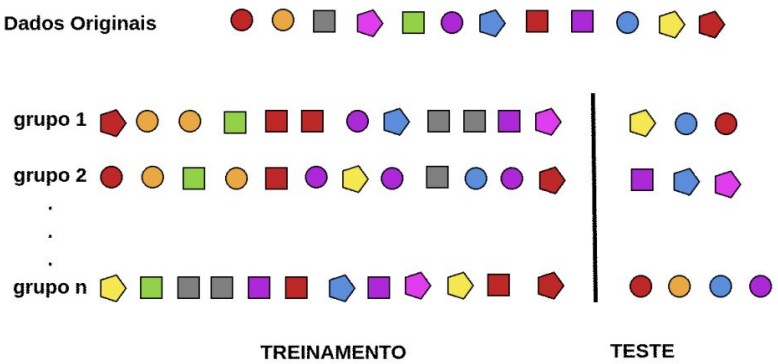


Classificação - Conjuntos de Treinamento e Teste

Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **Bootstrap**: a partir do conjunto original de dados, são gerados um número de conjuntos de amostras (com o mesmo tamanho do conjunto original) que serão usados no treinamento.

  Para um dado conjunto amostrados, podemos ter mais de uma cópia de alguns elementos. Aqueles elementos do conjunto original que não aparecem no conjunto de treinamento são usados na avaliação do modelo.



Classificação - Conjuntos de Treinamento e Teste

Técnicas de reamostragem: o processo de seleção de amostras para o conjunto de treinamento e para o conjunto de teste é repetido várias vezes.

- **Bootstrap**: a partir do conjunto original de dados, são gerados um número de conjuntos de amostras (com o mesmo tamanho do conjunto original) que serão usados no treinamento.

  Para um dado conjunto amostrados, podemos ter mais de uma cópia de alguns elementos. Aqueles elementos do conjunto original que não aparecem no conjunto de treinamento são usados na avaliação do modelo.

- No caso de um dataset muito grande, pode-se considerar usar um tamanho menor de dados (50%-80%).

- Deve-se fazer um número de repetições grande que permita uma estatística significativa (mímino 30 repetições).