CE092- Extensões do modelo de regressão

Cesar Augusto Taconeli

17 de novembro, 2026

O que vamos estudar nesta aula

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.

- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.
- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.
- **Vantagens e limitações do CART:** interpretabilidade, flexibilidade, subajuste e sobreajuste.

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.
- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.
- **Vantagens e limitações do CART:** interpretabilidade, flexibilidade, subajuste e sobreajuste.
- **Extensões do CART:**

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.
- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.
- **Vantagens e limitações do CART:** interpretabilidade, flexibilidade, subajuste e sobreajuste.
- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.
- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.
- **Vantagens e limitações do CART:** interpretabilidade, flexibilidade, subajuste e sobreajuste.
- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;
  - **Conditional Inference Trees (CIT):** seleção de variáveis baseada em testes estatísticos;

O que vamos estudar nesta aula

- **Árvores de classificação e regressão (CART):** princípios de funcionamento, regras de partição e critérios de impureza.
- **Construção e seleção de árvores:** ramificação, poda, custo-complexidade e validação cruzada.
- **Vantagens e limitações do CART:** interpretabilidade, flexibilidade, subajuste e sobreajuste.
- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;
  - **Conditional Inference Trees (CIT):** seleção de variáveis baseada em testes estatísticos;
  - **Model-Based Recursive Partitioning (MOB):** árvores com modelos de regressão nos nós terminais.

Introdução

Árvores de classificação e regressão

- Árvores de classificação e regressão (*Classification and Regression Trees — CART*) são métodos não paramétricos que permitem modelar respostas numéricas ou categóricas.

Árvores de classificação e regressão

- Árvores de classificação e regressão (*Classification and Regression Trees — CART*) são métodos não paramétricos que permitem modelar respostas numéricas ou categóricas.

- Permitem investigar a relação entre uma variável resposta e um conjunto de variáveis explicativas.

Árvores de classificação e regressão

- Árvores de classificação e regressão (*Classification and Regression Trees — CART*) são métodos não paramétricos que permitem modelar respostas numéricas ou categóricas.

- Permitem investigar a relação entre uma variável resposta e um conjunto de variáveis explicativas.

- Quando a resposta é numérica, temos uma **árvore de regressão**; quando é categórica, uma **árvore de classificação**.

Como funciona a técnica?

- O CART divide sucessivamente a amostra em duas subamostras, formando uma **estrutura de partições binárias**.

Como funciona a técnica?

- O CART divide sucessivamente a amostra em duas subamostras, formando uma **estrutura de partições binárias**.

- Cada divisão é determinada por uma **regra de partição**, baseada nos valores de um dos preditores.

Como funciona a técnica?

- O CART divide sucessivamente a amostra em duas subamostras, formando uma **estrutura de partições binárias**.
- Cada divisão é determinada por uma **regra de partição**, baseada nos valores de um dos preditores.
- A cada etapa, procura-se a divisão que proporcione a **maior redução da heterogeneidade** da variável resposta.

Como funciona a técnica?

- O CART divide sucessivamente a amostra em duas subamostras, formando uma **estrutura de partições binárias**.
- Cada divisão é determinada por uma **regra de partição**, baseada nos valores de um dos preditores.
- A cada etapa, procura-se a divisão que proporcione a **maior redução da heterogeneidade** da variável resposta.
- O processo é repetido recursivamente, até que algum critério de parada seja atendido.

Por que *árvore*?

- O conjunto de partições e regras de decisão pode ser representado graficamente por uma estrutura de árvore.

Por que *árvore*?

- O conjunto de partições e regras de decisão pode ser representado graficamente por uma estrutura de árvore.
  - O **nó raiz** representa a amostra original.

Por que *árvore*?

- O conjunto de partições e regras de decisão pode ser representado graficamente por uma estrutura de árvore.
  - O **nó raiz** representa a amostra original.
  - Os **nós internos** representam subconjuntos de observações submetidos a novas partições.

Por que *árvore*?

- O conjunto de partições e regras de decisão pode ser representado graficamente por uma estrutura de árvore.
  - O **nó raiz** representa a amostra original.
  - Os **nós internos** representam subconjuntos de observações submetidos a novas partições.
  - Os **nós terminais** (folhas) representam os grupos finais, nos quais são realizadas as predições.

Por que *árvore*?

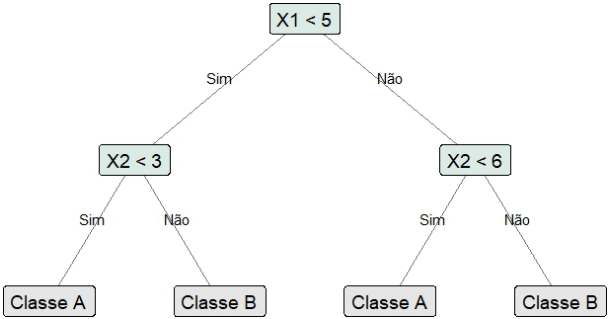


Figura 1: Ilustração de uma árvore de classificação.

Por que *árvore*?

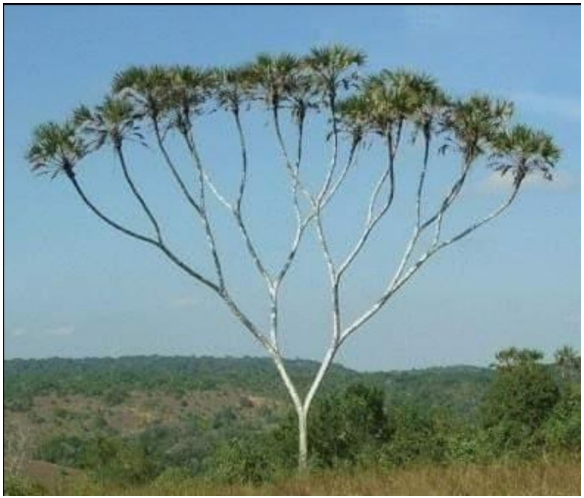


Figura 2: Motivação para o termo *árvore*.

Etapas da construção de uma árvore de classificação ou regressão

1. **Ramificação:** Construção da árvore por meio de partições binárias recursivas, respeitando critérios de parada.

Etapas da construção de uma árvore de classificação ou regressão

1. **Ramificação:** Construção da árvore por meio de partições binárias recursivas, respeitando critérios de parada.

2. **Poda:** Eliminação sucessiva de divisões, produzindo uma sequência de subárvores de complexidade decrescente.

Etapas da construção de uma árvore de classificação ou regressão

1. **Ramificação:** Construção da árvore por meio de partições binárias recursivas, respeitando critérios de parada.

2. **Poda:** Eliminação sucessiva de divisões, produzindo uma sequência de subárvores de complexidade decrescente.

3. **Seleção:** Escolha do tamanho da árvore, geralmente com base no erro preditivo estimado por validação cruzada.

Etapas da construção de uma árvore de classificação ou regressão

1. **Ramificação:** Construção da árvore por meio de partições binárias recursivas, respeitando critérios de parada.

2. **Poda:** Eliminação sucessiva de divisões, produzindo uma sequência de subárvores de complexidade decrescente.

3. **Seleção:** Escolha do tamanho da árvore, geralmente com base no erro preditivo estimado por validação cruzada.

4. **Interpretação:** Análise das regras de decisão e utilização da árvore selecionada para fins exploratórios ou preditivos.

Exemplo 1 - Concessão de crédito

Exemplo 1 - Concessão de crédito

- Nesta aplicação, vamos analisar dados da base `german.credit`, disponível no pacote `fairml` do `R`.

Exemplo 1 - Concessão de crédito

- Nesta aplicação, vamos analisar dados da base `german.credit`, disponível no pacote `fairml` do `R`.

- A base contém informações sobre 1.000 operações de crédito na Alemanha, das quais 700 foram classificadas como de bom risco e 300 como de mau risco.

Exemplo 1 - Concessão de crédito

- Nesta aplicação, vamos analisar dados da base `german.credit`, disponível no pacote `fairml` do `R`.

- A base contém informações sobre 1.000 operações de crédito na Alemanha, das quais 700 foram classificadas como de bom risco e 300 como de mau risco.

- O objetivo é classificar as operações de crédito como de bom ou mau risco, utilizando árvores de classificação (CART) e considerando as características dos clientes e das operações como preditores.

Exemplo 1 - Concessão de crédito

- As variáveis analisadas são as seguintes:

Exemplo 1 - Concessão de crédito

- As variáveis analisadas são as seguintes:
  - `Credit_risk`: classificação do risco de crédito (`BAD`: mau risco; `GOOD`: bom risco), variável resposta;

Exemplo 1 - Concessão de crédito

- As variáveis analisadas são as seguintes:
  - `Credit_risk`: classificação do risco de crédito (`BAD`: mau risco; `GOOD`: bom risco), variável resposta;
  - `Account_status`: situação da conta corrente;
  - `Duration`: duração do empréstimo (meses);
  - `Credit_history`: histórico de crédito;
  - `Purpose`: finalidade do empréstimo;
  - `Credit_amount`: valor do empréstimo;
  - `Savings_bonds`: recursos disponíveis em poupança e investimentos;
  - `Present_employment_since`: tempo no emprego atual;
  - `Installment_rate`: percentual da renda disponível comprometido com as parcelas;
  - `Other_debtors_guarantors`: existência de outros devedores ou garantidores;
  - `Resident_since`: tempo de residência no endereço atual (anos);

Exemplo 1 - Concessão de crédito

- Variáveis analisadas (continuação):
  - `Property`: tipo de propriedade ou patrimônio;
  - `Age`: idade do cliente (anos);
  - `Other_installment_plans`: existência de outros planos de pagamento;
  - `Housing`: condição de moradia;
  - `Existing_credits`: número de créditos existentes no banco;
  - `Job`: categoria ocupacional;
  - `People_maintenance_for`: número de pessoas dependentes financeiramente;
  - `Telephone`: disponibilidade de telefone;
  - `Foreign_worker`: condição de trabalhador estrangeiro;
  - `Gender`: gênero do cliente.

Ramificação da árvore

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;
  - Renda per capita $\leq$ 2 s.m. vs renda per capita $>$ 2 s.m.;

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;
  - Renda per capita $\leq$ 2 s.m. vs renda per capita $>$ 2 s.m.;
  - Sexo masculino vs sexo feminino;

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;
  - Renda per capita $\leq$ 2 s.m. vs renda per capita > 2 s.m.;
  - Sexo masculino vs sexo feminino;
  - Estado civil solteiro vs estado civil casado, viúvo ou divorciado;

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;
  - Renda per capita $\leq$ 2 s.m. vs renda per capita > 2 s.m.;
  - Sexo masculino vs sexo feminino;
  - Estado civil solteiro vs estado civil casado, viúvo ou divorciado;
  - Estado civil solteiro ou divorciado vs estado civil casado ou viúvo.

Regras de partição

- As regras de partição baseiam-se nas variáveis explicativas e em seus valores amostrados.

- Exemplos de regras de partição:
  - Idade <= 30 anos vs idade > 30 anos;
  - Renda per capita $\leq$ 2 s.m. vs renda per capita > 2 s.m.;
  - Sexo masculino vs sexo feminino;
  - Estado civil solteiro vs estado civil casado, viúvo ou divorciado;
  - Estado civil solteiro ou divorciado vs estado civil casado ou viúvo.

- Para preditores numéricos, essas partições produzem regiões (hiper)retangulares do espaço dos preditores.

Árvores de classificação e regressão

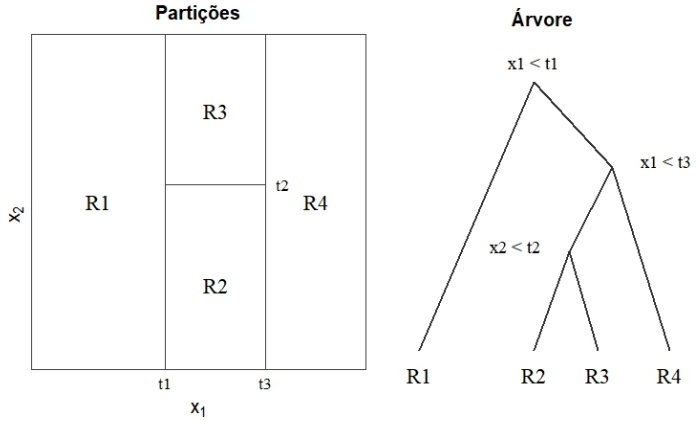


Figura 3: Partições e árvore de classificação ou regressão

Qual regra de partição aplicar?

- Em cada nó, existem diversas regras candidatas de partição, definidas a partir dos preditores.

Qual regra de partição aplicar?

- Em cada nó, existem diversas regras candidatas de partição, definidas a partir dos preditores.

- O objetivo é selecionar a regra que proporcione a maior redução da impureza da resposta.

Qual regra de partição aplicar?

- Em cada nó, existem diversas regras candidatas de partição, definidas a partir dos preditores.
- O objetivo é selecionar a regra que proporcione a maior redução da impureza da resposta.
- Para isso, comparamos a impureza do nó original com a média ponderada das impurezas dos dois nós resultantes.

Qual regra de partição aplicar?

- Em cada nó, existem diversas regras candidatas de partição, definidas a partir dos preditores.
- O objetivo é selecionar a regra que proporcione a maior redução da impureza da resposta.
- Para isso, comparamos a impureza do nó original com a média ponderada das impurezas dos dois nós resultantes.
- A escolha depende de uma medida de impureza, como o índice de Gini, a entropia ou a soma de quadrados de resíduos.

Qual regra de partição aplicar?

- Seja $I(t)$ a impureza de um nó original t e $I(t_L)$ e $I(t_R)$ as impurezas de dois nós filhos t_L e t_R.

Qual regra de partição aplicar?

- Seja $I(t)$ a impureza de um nó original t e $I(t_L)$ e $I(t_R)$ as impurezas de dois nós filhos t_L e t_R.

- A melhor partição é aquela que maximiza a redução da impureza (ΔI):

$$\Delta I = I(t) - \left[\frac{n_L}{n_t} I(t_L) + \frac{n_R}{n_t} I(t_R)\right],$$

em que n_t, n_L e n_R são os tamanhos de amostra em cada nó.

Qual regra de partição aplicar?

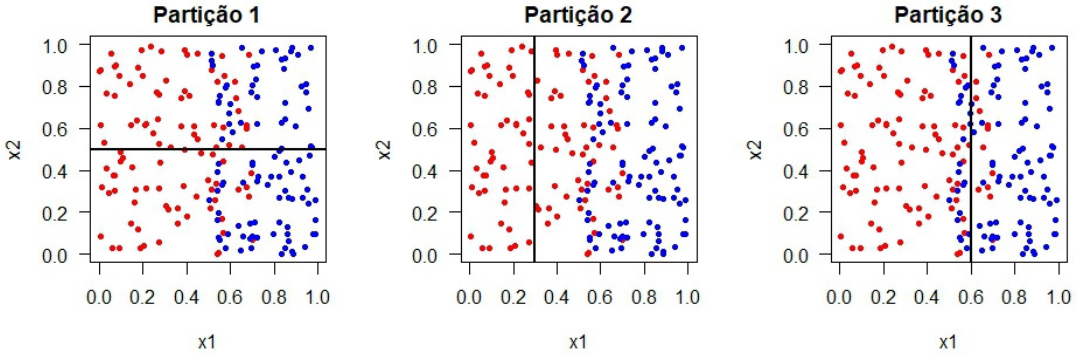


Figura 4: Árvore de classificação com diferentes partições (pontos azuis e vermelhos representam categorias diferentes para a resposta)

Qual regra de partição aplicar?

- Na Figura 4, são apresentadas três partições candidatas:

Qual regra de partição aplicar?

- Na Figura 4, são apresentadas três partições candidatas:
  - **Partição 1:** utiliza X_2, mas proporciona menor separação entre as categorias da resposta.

Qual regra de partição aplicar?

- Na Figura 4, são apresentadas três partições candidatas:
  - **Partição 1:** utiliza X_2, mas proporciona menor separação entre as categorias da resposta.
  - **Partição 2:** utiliza X_1, mas o ponto de corte não proporciona a melhor separação.

Qual regra de partição aplicar?

- Na Figura 4, são apresentadas três partições candidatas:
  - **Partição 1:** utiliza X_2, mas proporciona menor separação entre as categorias da resposta.
  - **Partição 2:** utiliza X_1, mas o ponto de corte não proporciona a melhor separação.
  - **Partição 3:** utiliza X_1 com um ponto de corte mais adequado, produzindo nós mais homogêneos.

Qual regra de partição aplicar?

- Na Figura 4, são apresentadas três partições candidatas:
  - **Partição 1:** utiliza X_2, mas proporciona menor separação entre as categorias da resposta.
  - **Partição 2:** utiliza X_1, mas o ponto de corte não proporciona a melhor separação.
  - **Partição 3:** utiliza X_1 com um ponto de corte mais adequado, produzindo nós mais homogêneos.

- Entre as três alternativas, a partição 3 apresenta a maior redução da impureza.

Exemplo 2 - Iris dataset (Ronald Fisher)

Exemplo 2 - Iris dataset (Ronald Fisher)

- A base iris, utilizada por Ronald Fisher em seu trabalho sobre análise discriminante, contém informações sobre 150 flores de três espécies: setosa, versicolor e virginica.

Exemplo 2 - Iris dataset (Ronald Fisher)

- A base iris, utilizada por Ronald Fisher em seu trabalho sobre análise discriminante, contém informações sobre 150 flores de três espécies: setosa, versicolor e virginica.

- Para cada flor, foram registradas quatro medidas, em centímetros:

Exemplo 2 - Iris dataset (Ronald Fisher)

- A base iris, utilizada por Ronald Fisher em seu trabalho sobre análise discriminante, contém informações sobre 150 flores de três espécies: setosa, versicolor e virginica.

- Para cada flor, foram registradas quatro medidas, em centímetros:
  - Comprimento e largura da sépala;

Exemplo 2 - Iris dataset (Ronald Fisher)

- A base iris, utilizada por Ronald Fisher em seu trabalho sobre análise discriminante, contém informações sobre 150 flores de três espécies: setosa, versicolor e virginica.

- Para cada flor, foram registradas quatro medidas, em centímetros:
  - Comprimento e largura da sépala;
  - Comprimento e largura da pétala.

Exemplo 2 - Iris dataset (Ronald Fisher)

- A base iris, utilizada por Ronald Fisher em seu trabalho sobre análise discriminante, contém informações sobre 150 flores de três espécies: setosa, versicolor e virginica.

- Para cada flor, foram registradas quatro medidas, em centímetros:
  - Comprimento e largura da sépala;
  - Comprimento e largura da pétala.

- O objetivo é produzir uma árvore de classificação para predizer a espécie de uma flor com base nessas medidas.

Exemplo 2 - Iris dataset (Ronald Fisher)



Figura 5: Espécies de iris.

Exemplo 2 - Iris dataset (Ronald Fisher)

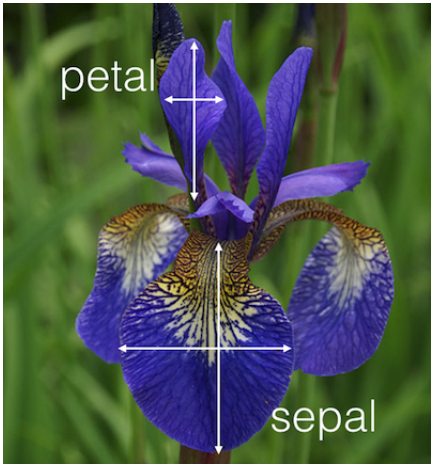


Figura 6: Variáveis usadas para classificação.

Exemplo 2 - Iris dataset (Ronald Fisher)

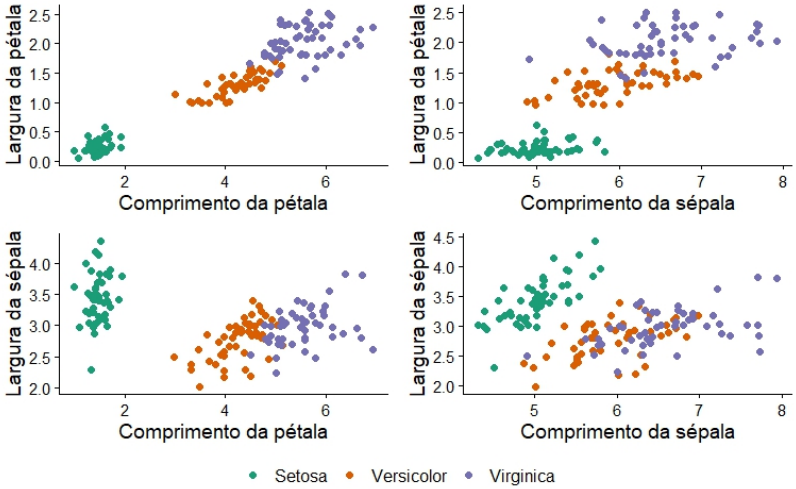


Figura 7: Dimensões das flores segundo a espécie.

Exemplo 2 - Iris dataset (Ronald Fisher)

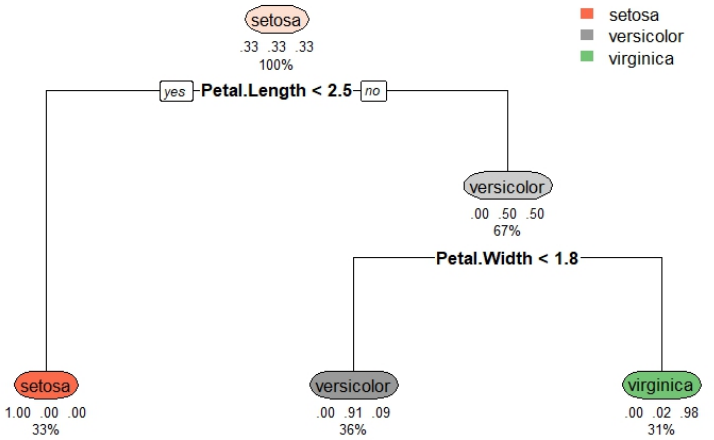


Figura 8: Árvore de classificação das espécies de iris.

Exemplo 2 - Iris dataset (Ronald Fisher)

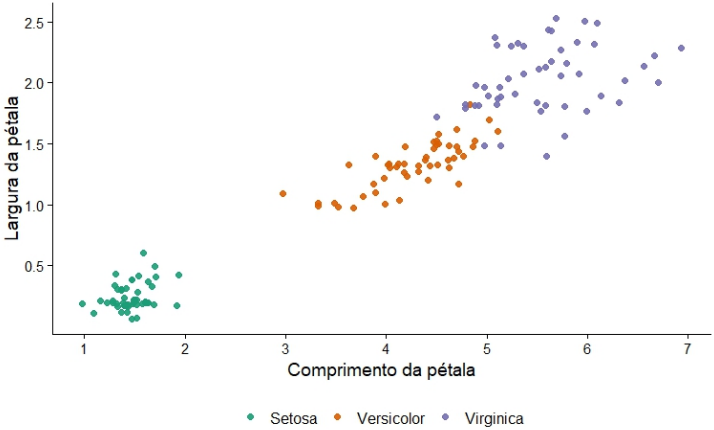


Figura 9: Dados de classificação de espécies de iris.

Exemplo 2 - Iris dataset (Ronald Fisher)

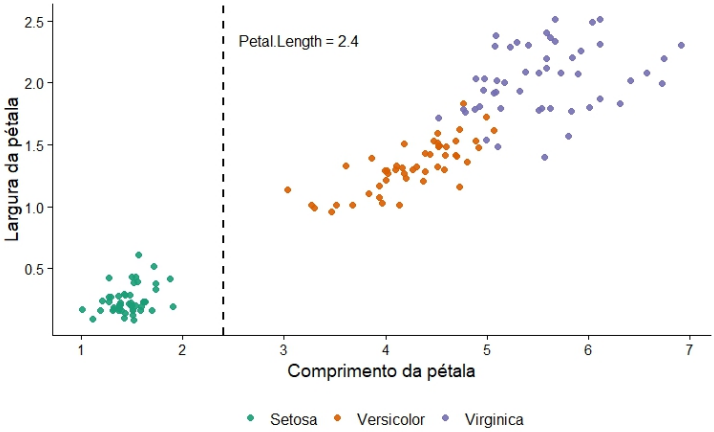


Figura 10: Dados de classificação de espécies de iris (primeira partição).

Exemplo 2 - Iris dataset (Ronald Fisher)

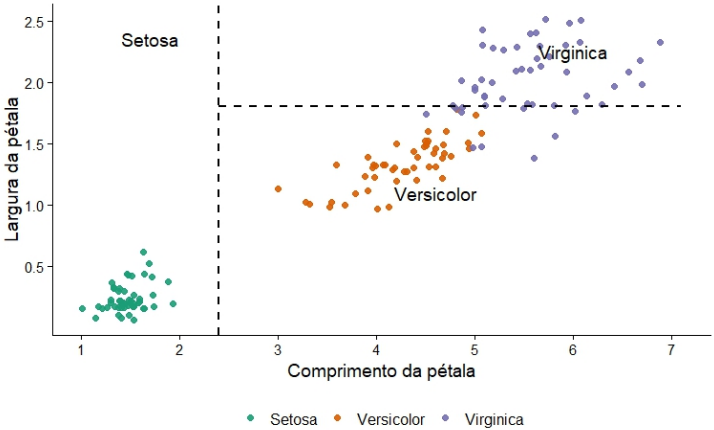


Figura 11: Dados de classificação de espécies de iris (partição final).

Exemplo 2 — Iris dataset

- As partições sucessivas dividem o espaço dos preditores em regiões cada vez mais homogêneas quanto à espécie.

Exemplo 2 — Iris dataset

- As partições sucessivas dividem o espaço dos preditores em regiões cada vez mais homogêneas quanto à espécie.

- Em cada nó terminal, a classe predita corresponde, usualmente, à espécie mais frequente entre as observações do nó.

Exemplo 2 — Iris dataset

- As partições sucessivas dividem o espaço dos preditores em regiões cada vez mais homogêneas quanto à espécie.
- Em cada nó terminal, a classe predita corresponde, usualmente, à espécie mais frequente entre as observações do nó.
- As proporções das espécies em cada nó também podem ser utilizadas como estimativas das probabilidades de classificação.

Exemplo 2 — Iris dataset

- As partições sucessivas dividem o espaço dos preditores em regiões cada vez mais homogêneas quanto à espécie.
- Em cada nó terminal, a classe predita corresponde, usualmente, à espécie mais frequente entre as observações do nó.
- As proporções das espécies em cada nó também podem ser utilizadas como estimativas das probabilidades de classificação.
- A capacidade preditiva da árvore deve ser avaliada com observações não utilizadas em seu ajuste.

Medidas de impureza

- Para árvores de regressão (resposta numérica), a medida de impureza mais usual é a **soma de quadrados dos resíduos (SQR)**.

$$\text{SQR}(t) = \sum_{i \in t} (y_i - \bar{y}_t)^2,$$

em que $\bar{y}_t$ representa a média da resposta entre as observações do nó t.

Medidas de impureza

- Para árvores de regressão (resposta numérica), a medida de impureza mais usual é a **soma de quadrados dos resíduos (SQR)**.

$$\text{SQR}(t) = \sum_{i \in t} (y_i - \bar{y}_t)^2,$$

em que $\bar{y}_t$ representa a média da resposta entre as observações do nó t.

- A média $\bar{y}_t$ é a predição que minimiza a SQR dentro do nó.

Medidas de impureza

- Para árvores de regressão (resposta numérica), a medida de impureza mais usual é a **soma de quadrados dos resíduos (SQR)**.

$$\text{SQR}(t) = \sum_{i \in t} (y_i - \bar{y}_t)^2,$$

em que $\bar{y}_t$ representa a média da resposta entre as observações do nó t.

- A média $\bar{y}_t$ é a predição que minimiza a SQR dentro do nó.

- A melhor partição é aquela que proporciona a maior redução na SQR, comparando o nó original com os dois nós resultantes.

Medidas de impureza

- Para árvores de classificação (resposta categórica), os **critérios de Gini e da entropia (ou informação)** são os mais usados.

Medidas de impureza

- Para árvores de classificação (resposta categórica), os **critérios de Gini e da entropia (ou informação)** são os mais usados.

- A título de ilustração, vamos considerar o caso em que a variável resposta tem duas categorias, e que as proporções de indivíduos pertencentes a cada uma delas em um nó t sejam p e $1 - p$.

Medidas de impureza

- Para árvores de classificação (resposta categórica), os **critérios de Gini e da entropia (ou informação)** são os mais usados.

- A título de ilustração, vamos considerar o caso em que a variável resposta tem duas categorias, e que as proporções de indivíduos pertencentes a cada uma delas em um nó t sejam p e $1 - p$.

- A impureza de t, medida pelos critérios da entropia e de Gini, fica definida por:

Medidas de impureza

- Para árvores de classificação (resposta categórica), os **critérios de Gini e da entropia (ou informação)** são os mais usados.

- A título de ilustração, vamos considerar o caso em que a variável resposta tem duas categorias, e que as proporções de indivíduos pertencentes a cada uma delas em um nó t sejam p e $1 - p$.

- A impureza de t, medida pelos critérios da entropia e de Gini, fica definida por:

$$\text{Entropia} = -p\ln(p) - (1-p)\ln(1-p)$$

$$\text{Gini} = 1 - p^2 - (1-p)^2 = 2p(1-p)$$

Medidas de impureza

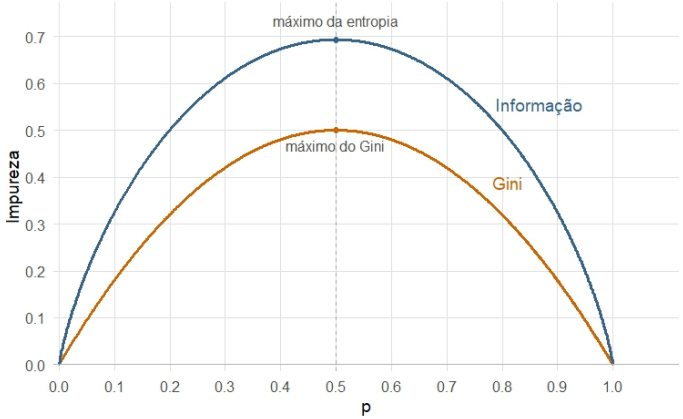


Figura 12: Medidas de impureza para árvores de classificação

Medidas de impureza

- Para ambas as medidas (entropia e Gini) temos:

Medidas de impureza

- Para ambas as medidas (entropia e Gini) temos:
  - $p = 0$ ou $p = 1$: nó puro, com impureza nula;

Medidas de impureza

- Para ambas as medidas (entropia e Gini) temos:
  - $p = 0$ ou $p = 1$: nó puro, com impureza nula;
  - $p = 0.5$: máxima impureza, pois as duas classes aparecem em proporções iguais;

Medidas de impureza

- Para ambas as medidas (entropia e Gini) temos:
  - $p = 0$ ou $p = 1$: nó puro, com impureza nula;
  - $p = 0.5$: máxima impureza, pois as duas classes aparecem em proporções iguais;
  - p próximo de 0 ou 1: baixa impureza.

Medidas de impureza

- Para ambas as medidas (entropia e Gini) temos:
  - $p = 0$ ou $p = 1$: nó puro, com impureza nula;
  - $p = 0.5$: máxima impureza, pois as duas classes aparecem em proporções iguais;
  - p próximo de 0 ou 1: baixa impureza.

- Na ramificação da árvore buscamos partições que reduzam ao máximo a impureza.

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

- Entre os principais critérios, destacam-se:

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

- Entre os principais critérios, destacam-se:
  - Número mínimo de observações em cada nó terminal;

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

- Entre os principais critérios, destacam-se:
  - Número mínimo de observações em cada nó terminal;
  - Número mínimo de observações em um nó para que uma nova partição seja considerada;

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

- Entre os principais critérios, destacam-se:
  - Número mínimo de observações em cada nó terminal;
  - Número mínimo de observações em um nó para que uma nova partição seja considerada;
  - Profundidade máxima da árvore;

Ramificação da árvore

- A árvore cresce por meio de partições recursivas, respeitando critérios de parada definidos pelo analista.

- Entre os principais critérios, destacam-se:
  - Número mínimo de observações em cada nó terminal;
  - Número mínimo de observações em um nó para que uma nova partição seja considerada;
  - Profundidade máxima da árvore;
  - Redução mínima de impureza necessária para realizar uma partição.

Poda da árvore

O processo de poda

- O processo de poda se inicia da árvore obtida na etapa anterior, após as sucessivas ramificações.

O processo de poda

- O processo de poda se inicia da árvore obtida na etapa anterior, após as sucessivas ramificações.

- A poda parte de uma árvore inicialmente construída e elimina sucessivamente algumas de suas ramificações.

O processo de poda

- O processo de poda se inicia da árvore obtida na etapa anterior, após as sucessivas ramificações.
- A poda parte de uma árvore inicialmente construída e elimina sucessivamente algumas de suas ramificações.
- O objetivo é obter árvores menos complexas, buscando preservar sua capacidade preditiva.

O processo de poda

- A **poda por custo-complexidade** baseia-se na minimização da função:

$$R_\alpha(T) = R(T) + \alpha|T|,$$

em que:

O processo de poda

- A **poda por custo-complexidade** baseia-se na minimização da função:

$$R_\alpha(T) = R(T) + \alpha|T|,$$

em que:

- $R(T)$ representa o custo associado ao ajuste da árvore;

O processo de poda

- A **poda por custo-complexidade** baseia-se na minimização da função:

$$R_\alpha(T) = R(T) + \alpha|T|,$$

em que:

- $R(T)$ representa o custo associado ao ajuste da árvore;
- $|T|$ é o número de nós terminais (folhas);

O processo de poda

- A **poda por custo-complexidade** baseia-se na minimização da função:

$$R_\alpha(T) = R(T) + \alpha|T|,$$

em que:

- $R(T)$ representa o custo associado ao ajuste da árvore;
- $|T|$ é o número de nós terminais (folhas);
- $\alpha \geq 0$ é o parâmetro que penaliza a complexidade.

O processo de poda

- O parâmetro α **controla a penalização** para a complexidade da árvore.

O processo de poda

- O parâmetro α **controla a penalização** para a complexidade da árvore.

- Para $\alpha = 0$, não há penalização da complexidade, favorecendo a maior árvore disponível.

O processo de poda

- O parâmetro α **controla a penalização** para a complexidade da árvore.
- Para $\alpha = 0$, não há penalização da complexidade, favorecendo a maior árvore disponível.
- À medida que α aumenta, árvores menores passam a ser favorecidas.

O processo de poda

- O parâmetro α **controla a penalização** para a complexidade da árvore.
- Para $\alpha = 0$, não há penalização da complexidade, favorecendo a maior árvore disponível.
- À medida que α aumenta, árvores menores passam a ser favorecidas.
- Para valores suficientemente elevados de α, a árvore selecionada contém apenas o nó raiz.

O processo de poda

- À medida que α aumenta, obtemos uma sequência de subárvores aninhadas, com complexidade decrescente.

O processo de poda

- À medida que α aumenta, obtemos uma sequência de subárvores aninhadas, com complexidade decrescente.

- Cada subárvore minimiza a função de custo-complexidade para determinados valores de α.

O processo de poda

- À medida que α aumenta, obtemos uma sequência de subárvores aninhadas, com complexidade decrescente.
- Cada subárvore minimiza a função de custo-complexidade para determinados valores de α.
- O procedimento permite avaliar o compromisso entre a qualidade do ajuste e o número de nós terminais.

O processo de poda

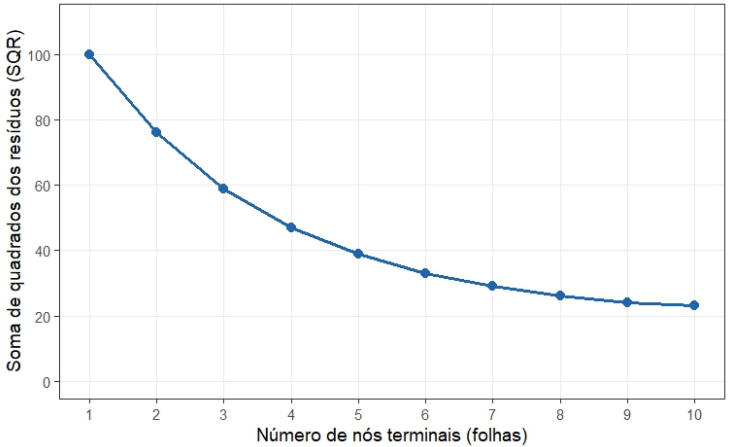


Figura 13: Ilustração de uma curva de custo-complexidade

Seleção da árvore

Qual o tamanho da árvore a ser selecionada?

- O tamanho de uma árvore é definido pelo **número de nós terminais**.

Qual o tamanho da árvore a ser selecionada?

- O tamanho de uma árvore é definido pelo **número de nós terminais**.
- Busca-se uma árvore relativamente simples, com bom desempenho preditivo em novas observações (**princípio da parcimônia**).

Qual o tamanho da árvore a ser selecionada?

- O tamanho de uma árvore é definido pelo **número de nós terminais**.
- Busca-se uma árvore relativamente simples, com bom desempenho preditivo em novas observações (**princípio da parcimônia**).
- Árvores muito pequenas: podem não capturar adequadamente as relações entre resposta e preditores (**underfitting**).

Qual o tamanho da árvore a ser selecionada?

- O tamanho de uma árvore é definido pelo **número de nós terminais**.
- Busca-se uma árvore relativamente simples, com bom desempenho preditivo em novas observações (**princípio da parcimônia**).
- Árvores muito pequenas: podem não capturar adequadamente as relações entre resposta e preditores (**underfitting**).
- Árvores muito grandes: podem se ajustar excessivamente às particularidades dos dados de treinamento (**overfitting**).

Qual o tamanho da árvore a ser selecionada?

- O tamanho de uma árvore é definido pelo **número de nós terminais**.
- Busca-se uma árvore relativamente simples, com bom desempenho preditivo em novas observações (**princípio da parcimônia**).
- Árvores muito pequenas: podem não capturar adequadamente as relações entre resposta e preditores (**underfitting**).
- Árvores muito grandes: podem se ajustar excessivamente às particularidades dos dados de treinamento (**overfitting**).
- O objetivo é encontrar um equilíbrio entre complexidade e capacidade de generalização.

Qual o tamanho da árvore a ser selecionada?

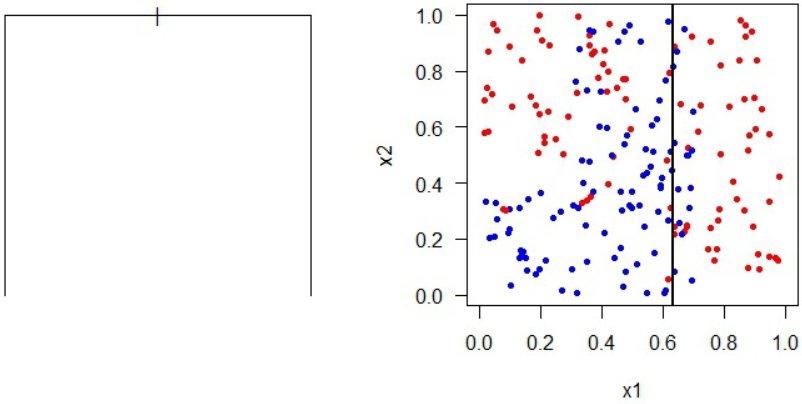


Figura 14: Árvore de classificação de tamanho insuficiente (*underfitting*).

Qual o tamanho da árvore a ser selecionada?

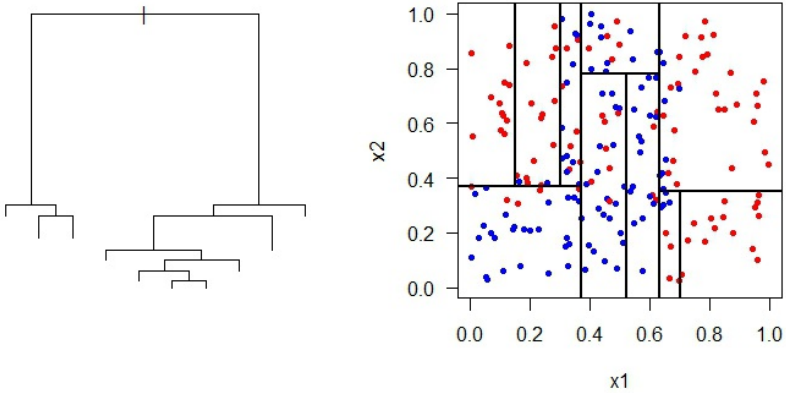


Figura 15: Árvore de classificação com excesso de ajuste (*overfitting*).

Qual o tamanho da árvore a ser selecionada?

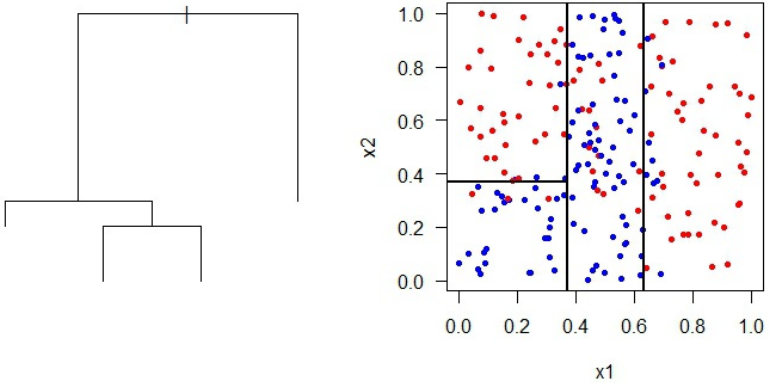


Figura 16: Árvore de classificação de tamanho adequado.

Seleção do modelo

- Após a poda, obtemos uma sequência de subárvores com diferentes níveis de complexidade.

Seleção do modelo

- Após a poda, obtemos uma sequência de subárvores com diferentes níveis de complexidade.

- O objetivo é selecionar uma árvore que apresente bom desempenho preditivo em novas observações.

Seleção do modelo

- Após a poda, obtemos uma sequência de subárvores com diferentes níveis de complexidade.
- O objetivo é selecionar uma árvore que apresente bom desempenho preditivo em novas observações.
- Para isso, podemos utilizar **validação cruzada**, estimando o erro preditivo das árvores com dados não utilizados em seus respectivos ajustes.

Seleção do modelo

- Após a poda, obtemos uma sequência de subárvores com diferentes níveis de complexidade.
- O objetivo é selecionar uma árvore que apresente bom desempenho preditivo em novas observações.
- Para isso, podemos utilizar **validação cruzada**, estimando o erro preditivo das árvores com dados não utilizados em seus respectivos ajustes.
- Esse procedimento auxilia na escolha da complexidade da árvore, reduzindo o risco de sobreajuste (overfitting).

Seleção do modelo

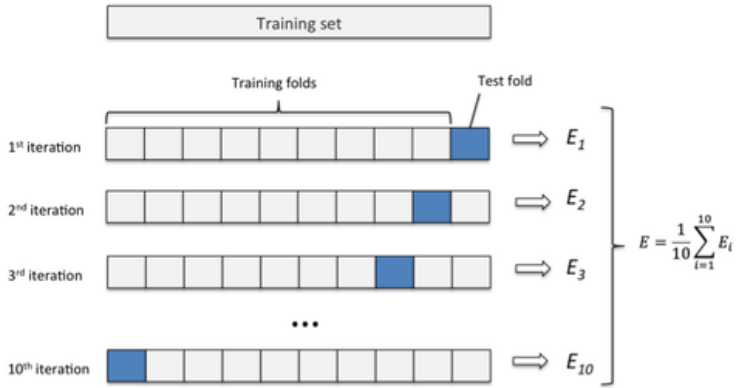


Figura 17: Ilustração da validação cruzada (10-fold).

Seleção do modelo

- Após a validação cruzada, podemos selecionar a árvore utilizando a regra de **um erro-padrão (1-SE)**:

Seleção do modelo

- Após a validação cruzada, podemos selecionar a árvore utilizando a regra de **um erro-padrão (1-SE)**:
  1. Identificar a árvore T^* com menor erro estimado por validação cruzada;

Seleção do modelo

- Após a validação cruzada, podemos selecionar a árvore utilizando a regra de **um erro-padrão (1-SE)**:
  1. Identificar a árvore T^* com menor erro estimado por validação cruzada;
  2. Calcular o erro-padrão (EP) associado à estimativa desse erro;

Seleção do modelo

- Após a validação cruzada, podemos selecionar a árvore utilizando a regra de **um erro-padrão (1-SE)**:
  1. Identificar a árvore T^* com menor erro estimado por validação cruzada;
  2. Calcular o erro-padrão (EP) associado à estimativa desse erro;
  3. Determinar o limite de tolerância:

$$L = \widehat{E}_{\mathrm{VC}}(T) + \mathrm{EP}(T);$$

Seleção do modelo

- Após a validação cruzada, podemos selecionar a árvore utilizando a regra de **um erro-padrão (1-SE)**:
  1. Identificar a árvore T^* com menor erro estimado por validação cruzada;
  2. Calcular o erro-padrão (EP) associado à estimativa desse erro;
  3. Determinar o limite de tolerância:

$$L = \widehat{E}_{\mathrm{VC}}(T) + \mathrm{EP}(T);$$

4. Selecionar **a menor árvore** cujo erro estimado por validação cruzada não ultrapasse L.

Seleção do modelo

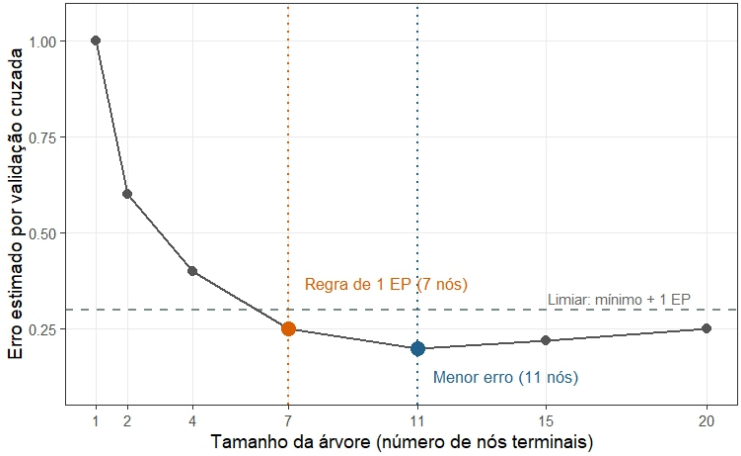


Figura 18: Curva de custo-complexidade produzida pela validação cruzada.

Interpretação da árvore

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.

- Cada nó terminal corresponde a um grupo de observações definido por uma sequência de regras.

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.

- Cada nó terminal corresponde a um grupo de observações definido por uma sequência de regras.

- Para árvores de classificação, podemos utilizar:

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.

- Cada nó terminal corresponde a um grupo de observações definido por uma sequência de regras.

- Para árvores de classificação, podemos utilizar:
  - **A classe mais frequente no nó**, para predizer a categoria da resposta;

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.

- Cada nó terminal corresponde a um grupo de observações definido por uma sequência de regras.

- Para árvores de classificação, podemos utilizar:
  - **A classe mais frequente no nó**, para predizer a categoria da resposta;
  - **As proporções das classes**, para estimar suas probabilidades.

Como interpretar a árvore resultante

- A interpretação de uma árvore baseia-se **nas regras de partição** e **nas características dos nós terminais**.
- Cada nó terminal corresponde a um grupo de observações definido por uma sequência de regras.
- Para árvores de classificação, podemos utilizar:
  - **A classe mais frequente no nó**, para predizer a categoria da resposta;
  - **As proporções das classes**, para estimar suas probabilidades.
- Para árvores de regressão, a predição usual é **a média da resposta** entre as observações do nó terminal.

Como ler a árvore

1. **Comece pelo nó raiz**, que representa a amostra original.

Como ler a árvore

1. **Comece pelo nó raiz**, que representa a amostra original.

2. **Siga as regras de partição**, escolhendo o ramo correspondente aos valores dos preditores.

Como ler a árvore

1. **Comece pelo nó raiz**, que representa a amostra original.

2. **Siga as regras de partição**, escolhendo o ramo correspondente aos valores dos preditores.

3. **Identifique o nó terminal** alcançado e o conjunto de regras que o define.

Como ler a árvore

1. **Comece pelo nó raiz**, que representa a amostra original.
2. **Siga as regras de partição**, escolhendo o ramo correspondente aos valores dos preditores.
3. **Identifique o nó terminal** alcançado e o conjunto de regras que o define.
4. **Obtenha a predição:** classe, probabilidade estimada ou média da resposta, conforme o problema.

Aspectos positivos e negativos do CART

Aspectos positivos

- Aplicável a problemas de classificação e regressão;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;
- Permite identificar grupos de observações com características semelhantes quanto à resposta;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;
- Permite identificar grupos de observações com características semelhantes quanto à resposta;
- É capaz de representar relações não lineares e interações entre preditores;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;
- Permite identificar grupos de observações com características semelhantes quanto à resposta;
- É capaz de representar relações não lineares e interações entre preditores;
- Não exige a especificação prévia de uma forma funcional para a relação entre resposta e preditores;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;
- Permite identificar grupos de observações com características semelhantes quanto à resposta;
- É capaz de representar relações não lineares e interações entre preditores;
- Não exige a especificação prévia de uma forma funcional para a relação entre resposta e preditores;
- Algumas implementações permitem lidar com dados ausentes por meio de regras alternativas de partição;

Aspectos positivos

- Aplicável a problemas de classificação e regressão;
- Produz regras de predição relativamente simples e interpretáveis;
- Permite identificar grupos de observações com características semelhantes quanto à resposta;
- É capaz de representar relações não lineares e interações entre preditores;
- Não exige a especificação prévia de uma forma funcional para a relação entre resposta e preditores;
- Algumas implementações permitem lidar com dados ausentes por meio de regras alternativas de partição;
- Pode ser aplicado a bases com numerosos preditores e observações.

Aspectos negativos

- Árvores individuais podem apresentar desempenho preditivo inferior ao de métodos mais flexíveis, como florestas aleatórias e métodos de boosting;

Aspectos negativos

- Árvores individuais podem apresentar desempenho preditivo inferior ao de métodos mais flexíveis, como florestas aleatórias e métodos de boosting;
- Podem ser instáveis: pequenas alterações nos dados podem produzir árvores bastante diferentes;

Aspectos negativos

- Árvores individuais podem apresentar desempenho preditivo inferior ao de métodos mais flexíveis, como florestas aleatórias e métodos de boosting;
- Podem ser instáveis: pequenas alterações nos dados podem produzir árvores bastante diferentes;
- Alguns algoritmos de construção apresentam viés na seleção de preditores com maior número de pontos de corte ou possibilidades de partição;

Aspectos negativos

- Árvores individuais podem apresentar desempenho preditivo inferior ao de métodos mais flexíveis, como florestas aleatórias e métodos de boosting;
- Podem ser instáveis: pequenas alterações nos dados podem produzir árvores bastante diferentes;
- Alguns algoritmos de construção apresentam viés na seleção de preditores com maior número de pontos de corte ou possibilidades de partição;
- Árvores muito grandes podem ser difíceis de interpretar;

Aspectos negativos

- Árvores individuais podem apresentar desempenho preditivo inferior ao de métodos mais flexíveis, como florestas aleatórias e métodos de boosting;
- Podem ser instáveis: pequenas alterações nos dados podem produzir árvores bastante diferentes;
- Alguns algoritmos de construção apresentam viés na seleção de preditores com maior número de pontos de corte ou possibilidades de partição;
- Árvores muito grandes podem ser difíceis de interpretar;
- As predições de árvores de regressão são constantes dentro de cada nó terminal, produzindo funções em degraus.

Exemplo 3 - Concentração de ozônio em Los Angeles

Exemplo 3 - Concentração de ozônio em Los Angeles

- Nesta aplicação, vamos analisar dados da base `ozone`, disponível no pacote `faraway` do R.

Exemplo 3 - Concentração de ozônio em Los Angeles

- Nesta aplicação, vamos analisar dados da base `ozone`, disponível no pacote `faraway` do R.

- A base contém **330 observações sobre a concentração de ozônio atmosférico e condições meteorológicas** na região de Los Angeles, Estados Unidos, em 1976.

Exemplo 3 - Concentração de ozônio em Los Angeles

- Nesta aplicação, vamos analisar dados da base `ozone`, disponível no pacote `faraway` do R.

- A base contém **330 observações sobre a concentração de ozônio atmosférico e condições meteorológicas** na região de Los Angeles, Estados Unidos, em 1976.

- O objetivo é **modelar a concentração de ozônio atmosférico em função das condições meteorológicas**, utilizando árvores de regressão (CART) para discriminar observações com diferentes concentrações médias de ozônio.

Exemplo 3 - Concentração de ozônio em Los Angeles

- As variáveis analisadas são as seguintes:

Exemplo 3 - Concentração de ozônio em Los Angeles

- As variáveis analisadas são as seguintes:
  - `O3`: concentração de ozônio atmosférico (ppm), variável resposta;

Exemplo 3 - Concentração de ozônio em Los Angeles

- As variáveis analisadas são as seguintes:
  - `O3`: concentração de ozônio atmosférico (ppm), variável resposta;
  - `vh`: medida meteorológica associada à altura de inversão;
  - `wind`: velocidade do vento;
  - `humidity`: umidade do ar;
  - `temp`: temperatura;
  - `ibh`: altura da base da inversão térmica;
  - `dpg`: gradiente de pressão atmosférica em Daggett;
  - `ibt`: temperatura da base da inversão térmica;
  - `vis`: visibilidade atmosférica;
  - `doy`: dia do ano (1 a 366).

Algumas extensões

Random forests

Random forests

- As **florestas aleatórias** (*random forests*) combinam um grande número de árvores de classificação ou regressão.

Random forests

- As **florestas aleatórias** (*random forests*) combinam um grande número de árvores de classificação ou regressão.

- Cada árvore é construída utilizando **uma amostra bootstrap dos dados originais**.

Random forests

- As **florestas aleatórias** (*random forests*) combinam um grande número de árvores de classificação ou regressão.
- Cada árvore é construída utilizando **uma amostra bootstrap dos dados originais**.
- Além disso, em cada partição, apenas **um subconjunto aleatório dos preditores é considerado**.

Random forests

- As **florestas aleatórias** (*random forests*) combinam um grande número de árvores de classificação ou regressão.
- Cada árvore é construída utilizando **uma amostra bootstrap dos dados originais**.
- Além disso, em cada partição, apenas **um subconjunto aleatório dos preditores é considerado**.
- **As predições das diferentes árvores são agregadas**, buscando melhorar a capacidade preditiva e reduzir a instabilidade de árvores individuais.

Random forests

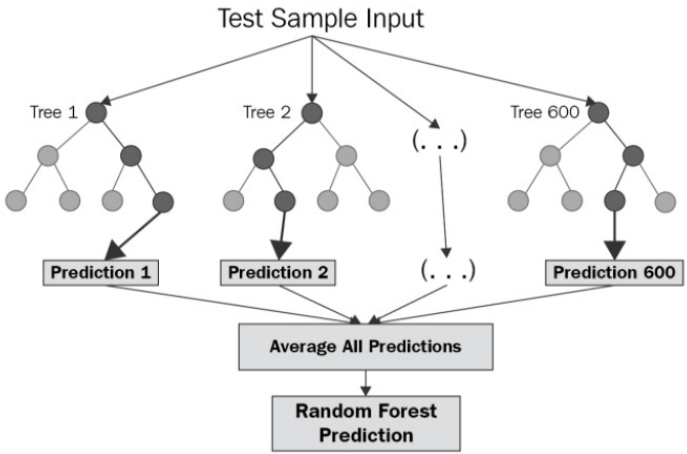


Figura 19: Random forest. Fonte: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:
  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:
  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;
  2. Construir uma árvore, considerando, em cada nó, um subconjunto aleatório de k preditores candidatos à partição;

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:
  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;
  2. Construir uma árvore, considerando, em cada nó, um subconjunto aleatório de k preditores candidatos à partição;
  3. Permitir o crescimento de uma árvore relativamente grande, geralmente sem poda;

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:
  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;
  2. Construir uma árvore, considerando, em cada nó, um subconjunto aleatório de k preditores candidatos à partição;
  3. Permitir o crescimento de uma árvore relativamente grande, geralmente sem poda;
  4. Repetir as etapas anteriores B vezes, produzindo B árvores;

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:

  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;
  2. Construir uma árvore, considerando, em cada nó, um subconjunto aleatório de k preditores candidatos à partição;
  3. Permitir o crescimento de uma árvore relativamente grande, geralmente sem poda;
  4. Repetir as etapas anteriores B vezes, produzindo B árvores;
  5. Agregar as predições: média para regressão e votação majoritária para classificação.

Random forests

- Considere uma amostra de n observações e p preditores. O algoritmo consiste nas seguintes etapas:
  1. Selecionar uma amostra bootstrap de tamanho n, com reposição, a partir dos dados originais;
  2. Construir uma árvore, considerando, em cada nó, um subconjunto aleatório de k preditores candidatos à partição;
  3. Permitir o crescimento de uma árvore relativamente grande, geralmente sem poda;
  4. Repetir as etapas anteriores B vezes, produzindo B árvores;
  5. Agregar as predições: média para regressão e votação majoritária para classificação.

- As observações não incluídas na amostra bootstrap de cada árvore (out-of-bag) podem ser utilizadas para estimar o erro preditivo.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- Nesta aplicação, vamos analisar dados da base `spam`, disponível no pacote `kernlab` do `R`.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- Nesta aplicação, vamos analisar dados da base `spam`, disponível no pacote `kernlab` do `R`.

- A base contém informações de 4.601 e-mails, dos quais 2.788 foram classificados como `não spam` e 1.813 como `spam`, coletados nos laboratórios da Hewlett-Packard.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- Nesta aplicação, vamos analisar dados da base `spam`, disponível no pacote `kernlab` do `R`.
- A base contém informações de 4.601 e-mails, dos quais 2.788 foram classificados como `não spam` e 1.813 como `spam`, coletados nos laboratórios da Hewlett-Packard.
- O objetivo é classificar os e-mails como spam ou não spam, utilizando florestas aleatórias (random forests) e considerando 57 características dos e-mails como preditores.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- Nesta aplicação, vamos analisar dados da base `spam`, disponível no pacote `kernlab` do `R`.
- A base contém informações de 4.601 e-mails, dos quais 2.788 foram classificados como `não spam` e 1.813 como `spam`, coletados nos laboratórios da Hewlett-Packard.
- O objetivo é classificar os e-mails como spam ou não spam, utilizando florestas aleatórias (random forests) e considerando 57 características dos e-mails como preditores.
- Os resultados serão comparados com os obtidos por uma árvore de classificação (CART), utilizando uma amostra independente para validação dos modelos.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- As variáveis analisadas são as seguintes:

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- As variáveis analisadas são as seguintes:
  - `type`: classificação do e-mail (`nonspam`: não spam; `spam`: spam), variável resposta.

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- As variáveis analisadas são as seguintes:
  - `type`: classificação do e-mail (`nonspam`: não spam; `spam`: spam), variável resposta.
  - 48 variáveis: frequências relativas de palavras específicas, como business, free e money;

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- As variáveis analisadas são as seguintes:
  - `type`: classificação do e-mail (`nonspam`: não spam; `spam`: spam), variável resposta.
  - 48 variáveis: frequências relativas de palavras específicas, como business, free e money;
  - 6 variáveis: frequências relativas de caracteres especiais, como ;, (, [, !, $ e #;

Exemplo 4 - Classificação de e-mails (spam vs. não spam)

- As variáveis analisadas são as seguintes:
  - `type`: classificação do e-mail (`nonspam`: não spam; `spam`: spam), variável resposta.
  - 48 variáveis: frequências relativas de palavras específicas, como business, free e money;
  - 6 variáveis: frequências relativas de caracteres especiais, como ;, (, [, !, $ e #;
  - 3 variáveis: características das sequências de letras maiúsculas (`capitalAve`, `capitalLong` e `capitalTotal`).

Conditional inference trees

Conditional inference trees

- Os algoritmos tradicionais de CART apresentam algumas características que motivaram o desenvolvimento de alternativas:

Conditional inference trees

- Os algoritmos tradicionais de CART apresentam algumas características que motivaram o desenvolvimento de alternativas:
  - As partições são selecionadas com base na redução da impureza, sem utilizar diretamente testes de significância estatística;

Conditional inference trees

- Os algoritmos tradicionais de CART apresentam algumas características que motivaram o desenvolvimento de alternativas:
  - As partições são selecionadas com base na redução da impureza, sem utilizar diretamente testes de significância estatística;
  - Preditores com maior número de partições candidatas podem ser favorecidos durante a seleção das divisões.

Conditional inference trees

- Os algoritmos tradicionais de CART apresentam algumas características que motivaram o desenvolvimento de alternativas:
  - As partições são selecionadas com base na redução da impureza, sem utilizar diretamente testes de significância estatística;
  - Preditores com maior número de partições candidatas podem ser favorecidos durante a seleção das divisões.

- As árvores de inferência condicional (conditional inference trees) utilizam testes estatísticos para selecionar os preditores associados à resposta.

Conditional inference trees

- Os algoritmos tradicionais de CART apresentam algumas características que motivaram o desenvolvimento de alternativas:
  - As partições são selecionadas com base na redução da impureza, sem utilizar diretamente testes de significância estatística;
  - Preditores com maior número de partições candidatas podem ser favorecidos durante a seleção das divisões.

- As árvores de inferência condicional (conditional inference trees) utilizam testes estatísticos para selecionar os preditores associados à resposta.

- Essa abordagem busca reduzir o viés de seleção de variáveis e utiliza a significância estatística como critério de parada.

Conditional inference trees

- O algoritmo `ctree` pode ser descrito, em linhas gerais, da seguinte forma:

Conditional inference trees

- O algoritmo `ctree` pode ser descrito, em linhas gerais, da seguinte forma:

1. **Seleção do preditor:** testar a hipótese nula global de independência entre a resposta e os preditores disponíveis no nó.

Conditional inference trees

- O algoritmo `ctree` pode ser descrito, em linhas gerais, da seguinte forma:

1. **Seleção do preditor:** testar a hipótese nula global de independência entre a resposta e os preditores disponíveis no nó.
   - Se a hipótese não for rejeitada, o processo de partição é encerrado;

Conditional inference trees

- O algoritmo `ctree` pode ser descrito, em linhas gerais, da seguinte forma:

1. **Seleção do preditor:** testar a hipótese nula global de independência entre a resposta e os preditores disponíveis no nó.
   - Se a hipótese não for rejeitada, o processo de partição é encerrado;
   - Caso contrário, selecionar o preditor com evidência estatística mais forte de associação com a resposta, considerando o ajuste para múltiplos testes.

2. **Escolha da partição:** para o preditor selecionado, identificar a divisão que maximiza a estatística de teste utilizada para avaliar a separação entre os grupos quanto à resposta.

2. **Escolha da partição:** para o preditor selecionado, identificar a divisão que maximiza a estatística de teste utilizada para avaliar a separação entre os grupos quanto à resposta.
   - **Preditores numéricos:** selecionar um ponto de corte;

2. **Escolha da partição:** para o preditor selecionado, identificar a divisão que maximiza a estatística de teste utilizada para avaliar a separação entre os grupos quanto à resposta.
   - **Preditores numéricos:** selecionar um ponto de corte;
   - **Preditores categóricos:** selecionar um agrupamento de categorias.

Conditional inference trees

2. **Escolha da partição:** para o preditor selecionado, identificar a divisão que maximiza a estatística de teste utilizada para avaliar a separação entre os grupos quanto à resposta.
   - **Preditores numéricos:** selecionar um ponto de corte;
   - **Preditores categóricos:** selecionar um agrupamento de categorias.

3. **Recursividade:** repetir o procedimento nos nós resultantes, até que os critérios de parada sejam atendidos.

Conditional inference trees

- A associação entre a resposta e os preditores é avaliada por meio de testes estatísticos baseados em permutação.

Conditional inference trees

- A associação entre a resposta e os preditores é avaliada por meio de testes estatísticos baseados em permutação.

- A abordagem permite considerar diferentes tipos de variáveis e estruturas de resposta, incluindo respostas numéricas, categóricas e de sobrevivência.

Conditional inference trees

- A associação entre a resposta e os preditores é avaliada por meio de testes estatísticos baseados em permutação.
- A abordagem permite considerar diferentes tipos de variáveis e estruturas de resposta, incluindo respostas numéricas, categóricas e de sobrevivência.
- Ajustes para múltiplos testes, como a correção de Bonferroni, podem ser utilizados na seleção dos preditores.

Conditional inference trees

- A associação entre a resposta e os preditores é avaliada por meio de testes estatísticos baseados em permutação.
- A abordagem permite considerar diferentes tipos de variáveis e estruturas de resposta, incluindo respostas numéricas, categóricas e de sobrevivência.
- Ajustes para múltiplos testes, como a correção de Bonferroni, podem ser utilizados na seleção dos preditores.
- Diferentemente da abordagem usual de CART, o `ctree` utiliza critérios estatísticos de parada, dispensando a etapa tradicional de poda por custo-complexidade.

Exemplo 5 - Classificação de tumores de mama

Exemplo 5 - Classificação de tumores de mama

- Nesta aplicação, vamos analisar dados da base `BreastCancer`, disponível no pacote `mlbench` do `R`.

Exemplo 5 - Classificação de tumores de mama

- Nesta aplicação, vamos analisar dados da base `BreastCancer`, disponível no pacote `mlbench` do `R`.

- A base contém informações de 699 registros de amostras citológicas mamárias, classificados como benignos ou malignos, a partir de características citológicas.

Exemplo 5 - Classificação de tumores de mama

- Nesta aplicação, vamos analisar dados da base `BreastCancer`, disponível no pacote `mlbench` do `R`.

- A base contém informações de 699 registros de amostras citológicas mamárias, classificados como benignos ou malignos, a partir de características citológicas.

- O objetivo é classificar os tumores como benignos ou malignos, utilizando árvores de inferência condicional (conditional inference trees) e considerando as nove características citológicas como preditores.

Exemplo 5 - Classificação de tumores de mama

- As variáveis analisadas são as seguintes:

Exemplo 5 - Classificação de tumores de mama

- As variáveis analisadas são as seguintes:
  - `Class:` classificação do tumor (benign: benigno; malignant: maligno);
  - `Cl.thickness:` espessura dos agrupamentos celulares;
  - `Cell.size:` uniformidade do tamanho das células;
  - `Cell.shape:` uniformidade da forma das células;
  - `Marg.adhesion:` adesão marginal das células;
  - `Epith.c.size:` tamanho das células epiteliais;
  - `Bare.nuclei:` presença de núcleos desnudos;
  - `Bl.cromatin:` características da cromatina;
  - `Normal.nucleoli:` características dos nucléolos;
  - `Mitoses:` atividade mitótica.

Exemplo 5 - Classificação de tumores de mama

- As variáveis analisadas são as seguintes:
  - `Class:` classificação do tumor (benign: benigno; malignant: maligno);
  - `Cl.thickness:` espessura dos agrupamentos celulares;
  - `Cell.size:` uniformidade do tamanho das células;
  - `Cell.shape:` uniformidade da forma das células;
  - `Marg.adhesion:` adesão marginal das células;
  - `Epith.c.size:` tamanho das células epiteliais;
  - `Bare.nuclei:` presença de núcleos desnudos;
  - `Bl.cromatin:` características da cromatina;
  - `Normal.nucleoli:` características dos nucléolos;
  - `Mitoses:` atividade mitótica.

- As nove características citológicas são registradas em escalas numéricas de 1 a 10.

Exemplo 6 - Recorrência de câncer de mama

Exemplo 6 - Recorrência de câncer de mama

- Nesta aplicação, vamos analisar dados do estudo German Breast Cancer Study Group 2 (GBSG2), disponíveis no pacote `TH.data` do R.

Exemplo 6 - Recorrência de câncer de mama

- Nesta aplicação, vamos analisar dados do estudo German Breast Cancer Study Group 2 (GBSG2), disponíveis no pacote `TH.data` do R.

- A base contém informações de 686 mulheres com câncer de mama, acompanhadas para avaliar o tempo livre de recorrência da doença.

Exemplo 6 - Recorrência de câncer de mama

- Nesta aplicação, vamos analisar dados do estudo German Breast Cancer Study Group 2 (GBSG2), disponíveis no pacote `TH.data` do R.

- A base contém informações de 686 mulheres com câncer de mama, acompanhadas para avaliar o tempo livre de recorrência da doença.

- O desfecho é o tempo até a ocorrência do evento de interesse, considerando a presença de observações censuradas.

Exemplo 6 - Recorrência de câncer de mama

- Nesta aplicação, vamos analisar dados do estudo German Breast Cancer Study Group 2 (GBSG2), disponíveis no pacote `TH.data` do R.

- A base contém informações de 686 mulheres com câncer de mama, acompanhadas para avaliar o tempo livre de recorrência da doença.

- O desfecho é o tempo até a ocorrência do evento de interesse, considerando a presença de observações censuradas.

- O objetivo é identificar grupos de pacientes com diferentes padrões de sobrevivência livre de recorrência, utilizando árvores de inferência condicional.

Exemplo 6 - Recorrência de câncer de mama

- As variáveis são as seguintes:

Exemplo 6 - Recorrência de câncer de mama

- As variáveis são as seguintes:
  - `time:` tempo livre de recorrência (dias);

Exemplo 6 - Recorrência de câncer de mama

- As variáveis são as seguintes:
  - `time`: tempo livre de recorrência (dias);
  - `cens`: indicador de ocorrência do evento (0: censura; 1: evento);

Exemplo 6 - Recorrência de câncer de mama

- As variáveis são as seguintes:
  - `time`: tempo livre de recorrência (dias);
  - `cens`: indicador de ocorrência do evento (0: censura; 1: evento);
  - `horTh`: terapia hormonal (no: não; yes: sim);
  - `age`: idade da paciente (anos);
  - `menostat`: estado menopausal (pre: pré-menopausa; post: pós-menopausa);
  - `tsize`: tamanho do tumor (mm);
  - `tgrade`: grau histológico do tumor (I, II ou III);
  - `pnodes`: número de linfonodos positivos;
  - `progrec`: nível de receptores de progesterona (fmol);
  - `estrec`: nível de receptores de estrogênio (fmol).

Model-Based Recursive Partitioning (Mob)

Model-Based Recursive Partitioning (Mob)

- O método MOB combina modelos de regressão paramétricos com particionamento recursivo.

Model-Based Recursive Partitioning (Mob)

- O método MOB combina modelos de regressão paramétricos com particionamento recursivo.

- Considere um modelo de regressão para a resposta Y, com preditores $x_1, x_2, \ldots, x_k$.

Model-Based Recursive Partitioning (Mob)

- O método MOB combina modelos de regressão paramétricos com particionamento recursivo.
- Considere um modelo de regressão para a resposta Y, com preditores $x_1, x_2, \ldots, x_k$.
- Adicionalmente, sejam $z_1, z_2, \ldots, z_q$ as variáveis candidatas ao particionamento.

Model-Based Recursive Partitioning (Mob)

- O método MOB combina modelos de regressão paramétricos com particionamento recursivo.
- Considere um modelo de regressão para a resposta Y, com preditores $x_1, x_2, \ldots, x_k$.
- Adicionalmente, sejam $z_1, z_2, \ldots, z_q$ as variáveis candidatas ao particionamento.
- O objetivo é identificar grupos de observações, definidos pelas variáveis $z_1, z_2, \ldots, z_q$, nos quais os parâmetros do modelo de regressão possam assumir valores diferentes.

Model-Based Recursive Partitioning (Mob)

- O método MOB combina modelos de regressão paramétricos com particionamento recursivo.
- Considere um modelo de regressão para a resposta Y, com preditores $x_1, x_2, \ldots, x_k$.
- Adicionalmente, sejam $z_1, z_2, \ldots, z_q$ as variáveis candidatas ao particionamento.
- O objetivo é identificar grupos de observações, definidos pelas variáveis $z_1, z_2, \ldots, z_q$, nos quais os parâmetros do modelo de regressão possam assumir valores diferentes.
- Em cada nó terminal, ajusta-se um modelo de regressão com a mesma estrutura, mas com parâmetros específicos daquele grupo.

Model-Based Recursive Partitioning (Mob)

- O algoritmo Mob é definido pelos seguintes passos:

Model-Based Recursive Partitioning (Mob)

- O algoritmo Mob é definido pelos seguintes passos:
  1. Ajustar o modelo de regressão utilizando as observações do nó considerado;

Model-Based Recursive Partitioning (Mob)

- O algoritmo Mob é definido pelos seguintes passos:
  1. Ajustar o modelo de regressão utilizando as observações do nó considerado;
  2. Avaliar a estabilidade dos parâmetros do modelo em relação às variáveis candidatas ao particionamento, por meio de testes estatísticos;

Model-Based Recursive Partitioning (Mob)

- O algoritmo Mob é definido pelos seguintes passos:
  1. Ajustar o modelo de regressão utilizando as observações do nó considerado;
  2. Avaliar a estabilidade dos parâmetros do modelo em relação às variáveis candidatas ao particionamento, por meio de testes estatísticos;
  3. Se houver evidências de instabilidade, selecionar uma variável e identificar a partição que proporciona o melhor ajuste dos modelos nos dois nós resultantes;

Model-Based Recursive Partitioning (Mob)

- O algoritmo Mob é definido pelos seguintes passos:
  1. Ajustar o modelo de regressão utilizando as observações do nó considerado;
  2. Avaliar a estabilidade dos parâmetros do modelo em relação às variáveis candidatas ao particionamento, por meio de testes estatísticos;
  3. Se houver evidências de instabilidade, selecionar uma variável e identificar a partição que proporciona o melhor ajuste dos modelos nos dois nós resultantes;
  4. Executar a partição e repetir o procedimento em cada nó formado, até que algum critério de parada seja atendido.

Exemplo 7 - Diagnóstico de diabetes

Exemplo 7 - Diagnóstico de diabetes

- Nesta aplicação, vamos analisar dados da base `PimaIndiansDiabetes`, disponível no pacote `mlbench` do `R`.

Exemplo 7 - Diagnóstico de diabetes

- Nesta aplicação, vamos analisar dados da base `PimaIndiansDiabetes`, disponível no pacote `mlbench` do `R`.

- A base contém informações de 768 mulheres indígenas Pima, com pelo menos 21 anos, incluindo características clínicas e o diagnóstico de diabetes.

Exemplo 7 - Diagnóstico de diabetes

- Nesta aplicação, vamos analisar dados da base `PimaIndiansDiabetes`, disponível no pacote `mlbench` do `R`.

- A base contém informações de 768 mulheres indígenas Pima, com pelo menos 21 anos, incluindo características clínicas e o diagnóstico de diabetes.

- O objetivo é modelar a probabilidade de diagnóstico de diabetes utilizando árvores de regressão baseadas em modelos (MOB).

Exemplo 7 - Diagnóstico de diabetes

- Nesta aplicação, vamos analisar dados da base `PimaIndiansDiabetes`, disponível no pacote `mlbench` do `R`.

- A base contém informações de 768 mulheres indígenas Pima, com pelo menos 21 anos, incluindo características clínicas e o diagnóstico de diabetes.

- O objetivo é modelar a probabilidade de diagnóstico de diabetes utilizando árvores de regressão baseadas em modelos (MOB).

Exemplo 7 - Diagnóstico de diabetes

- Em cada nó terminal, será ajustada uma regressão logística com a idade (age) como preditor:

$$\log\left(\frac{\pi_{\texttt{age}}}{1-\pi_{\texttt{age}}}\right) = \beta_0 + \beta_1 \times \texttt{age},$$

onde $\pi_{\texttt{age}}$ é a probabilidade de diagnóstico positivo para uma idade `age`.

Exemplo 7 - Diagnóstico de diabetes

- Em cada nó terminal, será ajustada uma regressão logística com a idade (age) como preditor:

$$\log\left(\frac{\pi_{\texttt{age}}}{1-\pi_{\texttt{age}}}\right) = \beta_0 + \beta_1 \times \texttt{age},$$

onde $\pi_{\texttt{age}}$ é a probabilidade de diagnóstico positivo para uma idade `age`.

- As demais características clínicas, exceto `age` e `glucose`, serão consideradas candidatas ao particionamento.

Exemplo 7 - Diagnóstico de diabetes

- **Variável resposta:**
  - `diabetes`: diagnóstico de diabetes (neg: não; pos: sim).

Exemplo 7 - Diagnóstico de diabetes

- **Variável resposta:**
  - `diabetes`: diagnóstico de diabetes (neg: não; pos: sim).
- **Preditor do modelo logístico:**
  - `age`: idade (anos).

Exemplo 7 - Diagnóstico de diabetes

- **Variável resposta:**
  - `diabetes`: diagnóstico de diabetes (neg: não; pos: sim).
- **Preditor do modelo logístico:**
  - `age`: idade (anos).
- **Variáveis candidatas ao particionamento:**
  - `pregnant`: número de gestações;
  - `pressure`: pressão arterial diastólica (mmHg);
  - `triceps`: espessura da dobra cutânea do tríceps (mm);
  - `insulin`: insulina sérica após duas horas (μU/mL);
  - `mass`: índice de massa corporal (kg/m^2);
  - `pedigree`: função de histórico familiar de diabetes.

Exemplo 8 - Valores de imóveis em Boston

Exemplo 8 - Valores de imóveis em Boston

- Nesta aplicação, vamos analisar dados da base `BostonHousing`, disponível no pacote `mlbench` do `R`.

Exemplo 8 - Valores de imóveis em Boston

- Nesta aplicação, vamos analisar dados da base `BostonHousing`, disponível no pacote `mlbench` do `R`.

- A base contém informações sobre 506 setores censitários da região de Boston, Estados Unidos, incluindo características socioeconômicas, habitacionais e ambientais.

Exemplo 8 - Valores de imóveis em Boston

- Nesta aplicação, vamos analisar dados da base `BostonHousing`, disponível no pacote `mlbench` do `R`.

- A base contém informações sobre 506 setores censitários da região de Boston, Estados Unidos, incluindo características socioeconômicas, habitacionais e ambientais.

- O objetivo é modelar o valor mediano dos imóveis (em milhares de dólares), utilizando árvores de regressão baseadas em modelos (MOB).

Exemplo 8 - Valores de imóveis em Boston

- Nesta aplicação, vamos analisar dados da base `BostonHousing`, disponível no pacote `mlbench` do `R`.

- A base contém informações sobre 506 setores censitários da região de Boston, Estados Unidos, incluindo características socioeconômicas, habitacionais e ambientais.

- O objetivo é modelar o valor mediano dos imóveis (em milhares de dólares), utilizando árvores de regressão baseadas em modelos (MOB).

Exemplo 8 - Valores de imóveis em Boston

- Em cada nó terminal, será ajustado um modelo de regressão linear com `log(lstat)` e `rm` como preditores:

$$\mu_{\mathrm{x}} = \beta_0 + \beta_1 \times \log(\texttt{lstat}) + \beta_2 \times \texttt{rm},$$

em que μ_{x} é o preço mediano esperado dado `lstat` e `rm`.

Exemplo 8 - Valores de imóveis em Boston

- Em cada nó terminal, será ajustado um modelo de regressão linear com `log(lstat)` e `rm` como preditores:

$$\mu_{\mathrm{x}} = \beta_0 + \beta_1 \times \log(\texttt{lstat}) + \beta_2 \times \texttt{rm},$$

em que μ_{x} é o preço mediano esperado dado `lstat` e `rm`.

- As demais características serão consideradas candidatas ao particionamento.

Exemplo 8 - Valores de imóveis em Boston

- **Variável resposta:**
  - `medv`: valor mediano dos imóveis, em milhares de dólares.

Exemplo 8 - Valores de imóveis em Boston

- **Variável resposta:**
  - `medv`: valor mediano dos imóveis, em milhares de dólares.

- **Preditores do modelo de regressão linear:**
  - `lstat`: percentual da população classificada como de menor status socioeconômico (utilizado em escala logarítmica);
  - `rm`: número médio de cômodos por residência.

Exemplo 8 - Valores de imóveis em Boston

- **Variável resposta:**
  - `medv`: valor mediano dos imóveis, em milhares de dólares.

- **Preditores do modelo de regressão linear:**
  - `lstat`: percentual da população classificada como de menor status socioeconômico (utilizado em escala logarítmica);
  - `rm`: número médio de cômodos por residência.

Exemplo 8 - Valores de imóveis em Boston

- **Variáveis candidatas ao particionamento:**

Exemplo 8 - Valores de imóveis em Boston

- **Variáveis candidatas ao particionamento:**
  - `crim`: taxa de criminalidade per capita;
  - `zn`: percentual de terrenos residenciais destinados a lotes com mais de 25.000 pés quadrados;
  - `indus`: proporção de áreas destinadas a atividades comerciais não varejistas;
  - `chas`: indicador de proximidade ao rio Charles;
  - `nox`: concentração de óxidos de nitrogênio;
  - `age`: proporção de unidades ocupadas por proprietários e construídas antes de 1940;
  - `dis`: distância ponderada até cinco centros de emprego;
  - `rad`: índice de acessibilidade às principais rodovias;
  - `tax`: taxa de imposto sobre propriedades por US$ 10.000;
  - `ptratio`: razão entre alunos e professores;
  - `b`: variável histórica derivada da composição racial da população.

O que estudamos nesta aula?

O que estudamos nesta aula?

- **Árvores de classificação e regressão (CART):** construção de regras de decisão por meio de partições recursivas dos dados.

O que estudamos nesta aula?

- **Árvores de classificação e regressão (CART):** construção de regras de decisão por meio de partições recursivas dos dados.

- **Construção e seleção de árvores:** critérios de impureza, poda por custo-complexidade e validação cruzada para controlar o sobreajuste.

O que estudamos nesta aula?

- **Árvores de classificação e regressão (CART):** construção de regras de decisão por meio de partições recursivas dos dados.
- **Construção e seleção de árvores:** critérios de impureza, poda por custo-complexidade e validação cruzada para controlar o sobreajuste.
- **Vantagens e limitações do CART:** interpretabilidade e flexibilidade, mas também instabilidade e risco de sobreajuste.

O que estudamos nesta aula?

- **Extensões do CART:**

O que estudamos nesta aula?

- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;

O que estudamos nesta aula?

- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;
  - **Conditional Inference Trees (CIT):** utilização de testes estatísticos para orientar a seleção de variáveis e as partições;

O que estudamos nesta aula?

- **Extensões do CART:**
  - **Random Forests:** combinação de árvores para melhorar a capacidade preditiva;
  - **Conditional Inference Trees (CIT):** utilização de testes estatísticos para orientar a seleção de variáveis e as partições;
  - **Model-Based Recursive Partitioning (MOB):** combinação de modelos de regressão com partições recursivas, permitindo parâmetros específicos para diferentes grupos.