Introdução a machine learning com aplicações em R

Aula 4 - Aprendizado não supervisionado e supervisionado

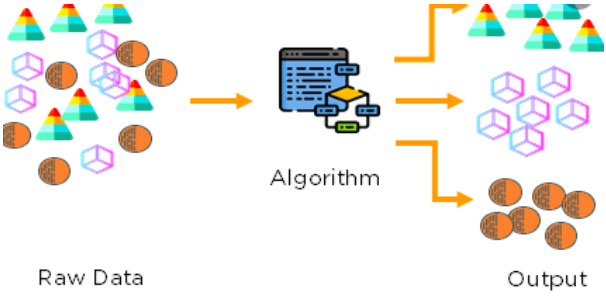
Felipe Barletta

Departamento de Estatística

10 novembro, 2020

Sumário

Aprendizado não supervisionado



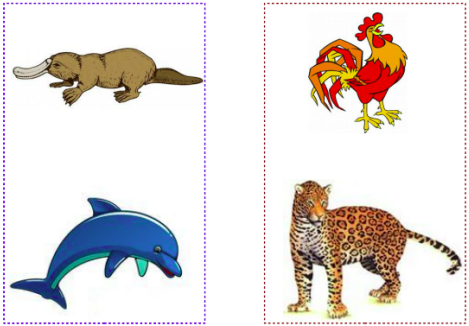
- Redução de dimensionalidade
- Agrupamento (*Clustering*)

Clustering

- *Clustering* refere-se ao conjunto de técnicas para encontrar subgrupos (ou clusters) a partir dos dados;
- Buscamos partições em grupos distintos, tal que observações dentro de cada grupo sejam similares entre si e diferentes dos demais;

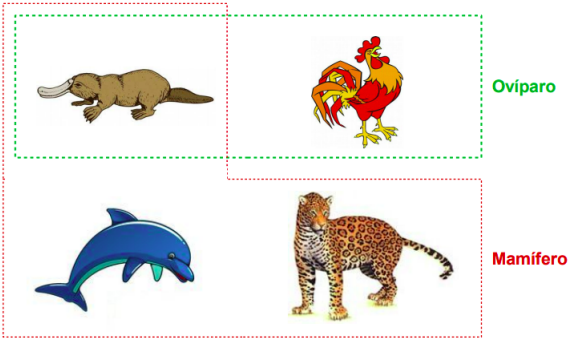
Clustering

- *Clustering* refere-se ao conjunto de técnicas para encontrar subgrupos (ou clusters) a partir dos dados;
- Buscamos partições em grupos distintos, tal que observações dentro de cada grupo sejam similares entre si e diferentes dos demais;



Clustering

- *Clustering* refere-se ao conjunto de técnicas para encontrar subgrupos (ou clusters) a partir dos dados;
- Buscamos partições em grupos distintos, tal que observações dentro de cada grupo sejam similares entre si e diferentes dos demais;



Clustering

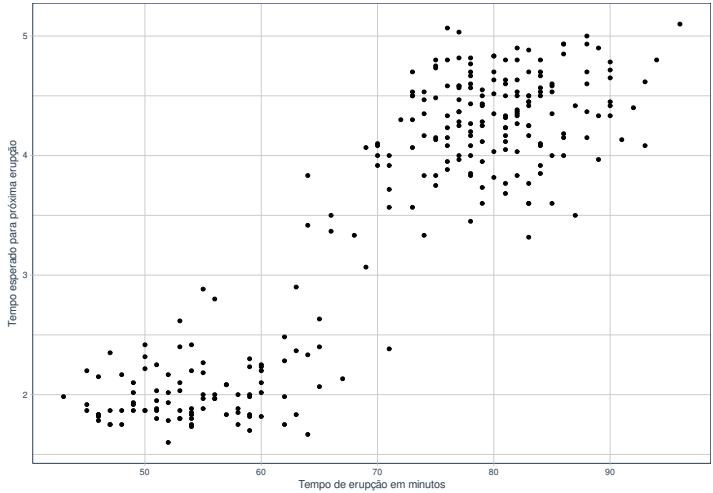
- *Clustering* refere-se ao conjunto de técnicas para encontrar subgrupos (ou clusters) a partir dos dados;
- Buscamos partições em grupos distintos, tal que observações dentro de cada grupo sejam similares entre si e diferentes dos demais;

1. Métodos não-hierárquicos
2. Métodos hierárquicos

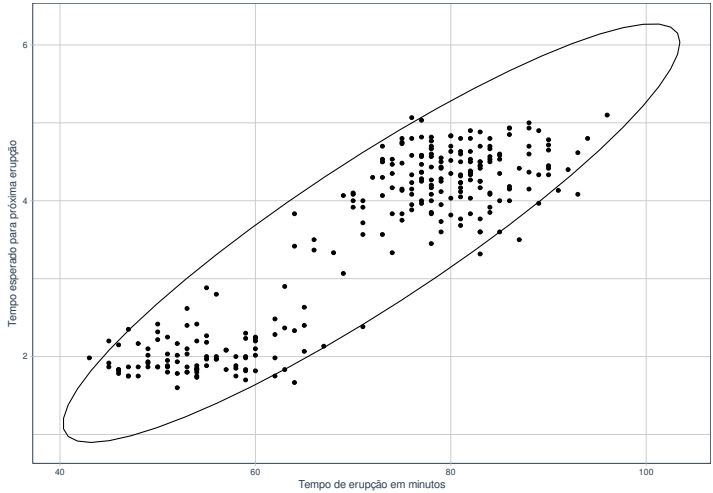
Clustering - método não-hierárquico

- A ideia central é escolher uma partição inicial dos elementos e, em seguida, alterar os membros dos grupos para obter-se o melhor particionamento (ANDERBERG, 1973).
- Quando comparado com o método hierárquico, este método é mais rápido, pois não é necessário calcular e armazenar, durante o processamento, a matriz de similaridade.
- *K-means* algoritmo não hierárquico

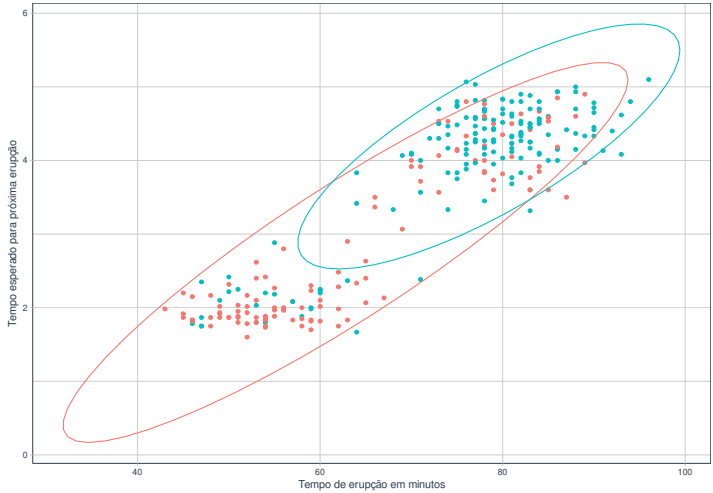
K-means


Tempo esperado para próxima erupção
Tempo de erupção em minutos
2
3
4
5
50
60
70
80
90

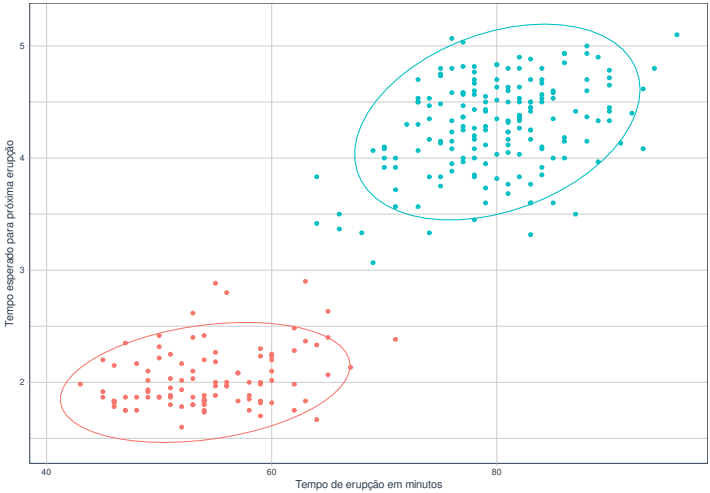
K-means


Tempo esperado para próxima erupção
6
4
2
40
60
80
100
Tempo de erupção em minutos

K-means


Tempo esperado para próxima erupção
Tempo de erupção em minutos
0
2
4
6
40
60
80
100

K-means


Tempo esperado para próxima erupção
5
4
3
2
40
60
80
Tempo de erupção em minutos

K-means

- Seja $C_1, C_2, ..., C_K$ respectivos grupos contendo os índices das observações, satisfazendo as seguintes propriedades:
  1. $C_1 \cup C_2 \cup ... \cup C_K = 1, ..., n$. Em outras palavras, cada observação pertence à pelo menos um grupo;
  2. $C_k \cap C_k = \emptyset$. Ou seja, as observações não pertencem a mais de um grupo ao mesmo tempo.
- A variação dentro do *cluster* C_k é medida por:

$$V(C_k) = \frac{1}{\mid C_K \mid} \sum_{i,i' \in C_K} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

em que $\mid C_k \mid$ denota o número de observações no k-ésimo *cluster*.

K-means

- Deseja-se

$$\underset{C_1,..,C_K}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \frac{1}{\mid C_K \mid} \sum_{i,i' \in C_K} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\} = \underset{C_1,..,C_K}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} 2 \sum_{i \in C_K} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2 \right\},$$

em que $\bar{x}_{kj} = \frac{1}{\mid C_K \mid} \sum_{i \in C_K} x_{ij}$

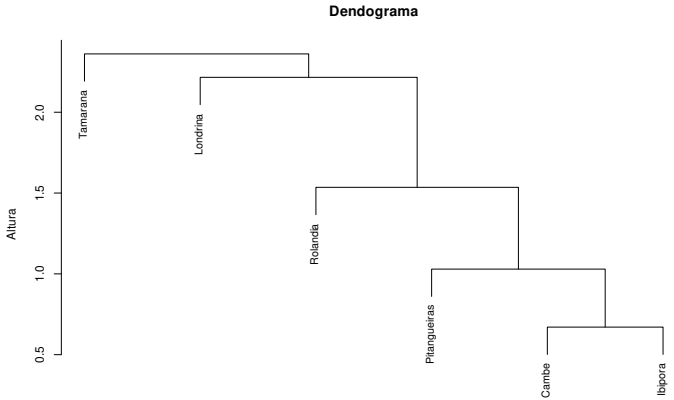
Algoritmo

- Passo 1: Atribua, aleatoriamente, cada observação em um dos K *clusters* ;
- Passo 2: Itere até que os *clusters* se estabilizem:
  a. Para cada K *cluster*, calcule seu centroide;
  b. Atribua cada observação ao cluster mais próximo (menor distância Euclideana).

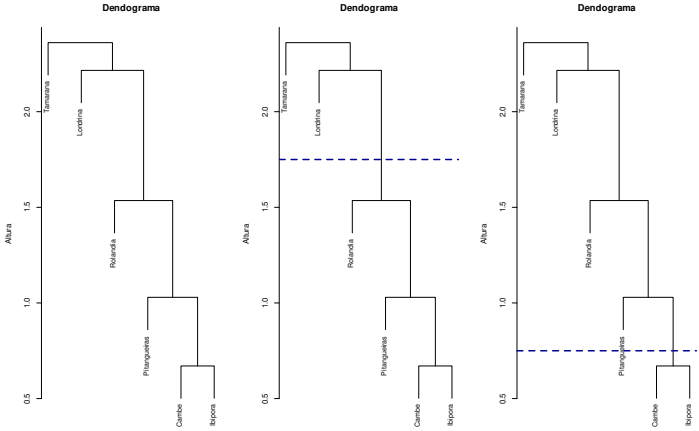
Clustering - método hierárquico

- É uma abordagem alternativa, que não exige a escolha de K;
- Iniciamos com cada ponto sendo seu próprio cluster;
- Identificamos os dois clusters mais próximos e os agrupamos;
- Repetimos este processo até restar um cluster.

Clustering - método hierárquico

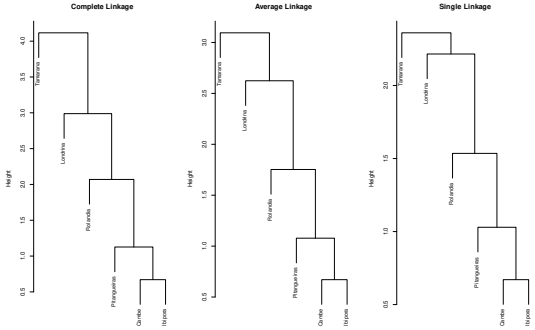


Clustering - método hierárquico

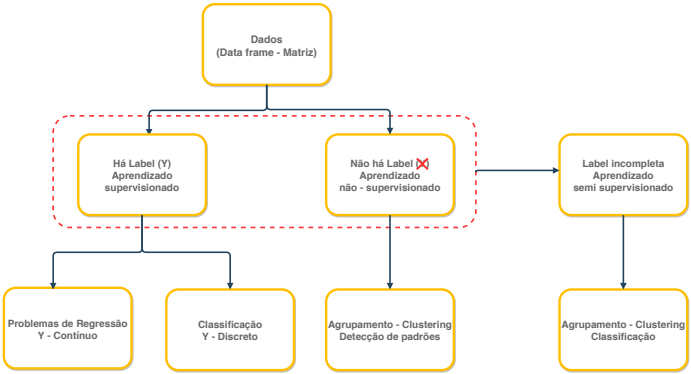


Clustering - Tipos de *linkage*

1. Complete - $L(r,s) = \max(D(x_{r,i}, ..., x_{s,j}))$
2. Single - $L(r,s) = \min(D(x_{r,i}, ..., x_{s,j}))$
3. Average - $L(r,s) = \dfrac{1}{n_r n_s} \sum_{i=1}^{nr} \sum_{j=1}^{ns} (D(x_{r,i}, ..., x_{s,j}))$



Aprendizado Supervisionado

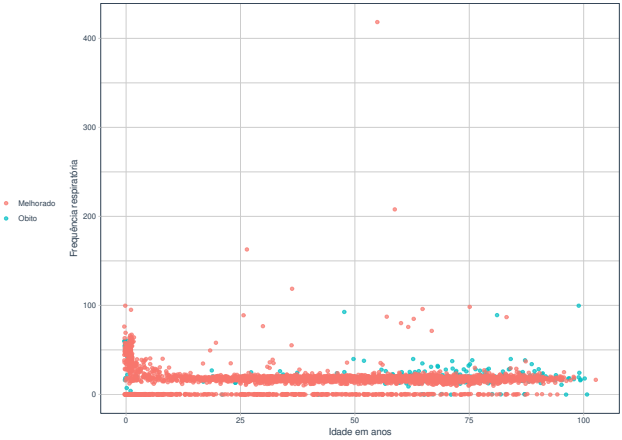


Classificação

- Em muitos problemas, a característica Y assume valores não ordenados em um conjunto C.
  a. Consumo de um produto $\in$ {sim, não};
  b. Sinistros $\in$ {fraude, não fraude}
  c. Paciente $\in$ {saudável, doente}
  d. Paciente em pronto atendimento $\in$ {Alta, Internação ou Óbito}
  e. Tipos de solos $\in \{S_1, S_2, S_3, ..., S_n\}$

Classificação

1. *Output* do modelo é uma probabilidade.
2. Estimamos $P(Y = c \mid x)$ para cada categoria $c \in C$.
3. Tomamos então $h(x) = \text{argmax}\, \hat{P}(Y = c \mid x)$



Regressão Logística

- Caso binário:

$$Y = \begin{cases} 0, & \text{se } \textit{Alta} \\ 1, & \text{se } \textit{bito} \end{cases}$$

- A regressão logística tem a seguinte a forma:

$$P(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X_1}}{1 + e^{\beta_0+\beta_1 X_1}}$$

- Com um pouco de algebrismo, chegamos em

$$\log\left[\frac{P(Y = 1|X)}{1 - P(Y = 1|X)}\right] = \beta_0 + \beta_1 X_1$$

Regressão Logística

- Com múltiplas variáveis:

$$\log\left[\frac{P(Y=1|X)}{1-P(Y=1|X)}\right] = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

- Em outras palavras:
  - Considere a problema binário onde Y assume valores, 0 ou 1
  - Para um dado vetor x, queremos estimar $P(Y = c \mid x)$

$$\beta_0 + \beta_1 X_1 + ... + \beta_p X_p \approx h(x)$$

Regressão Multinomial

- Generalizando para caso de Y com mais de duas classes:

$$P(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{l=1}^{K} e^{\beta_{0l}+\beta_{1l}X_1+\ldots+\beta_{pk}X_p}}$$

- k é o números de classes de Y
- Teremos $k-1$ *logitos*

Regressão Multinomial: Y em escala nominal

Exemplo: Desfecho de pacientes em um pronto atendimento de um hospital

$$Y = \begin{cases} 1, & \text{se } \textit{Alta} \\ 2, & \text{se } \textit{Internado} \\ 3, & \text{se } \textit{Obito} \end{cases}$$

$$\log\left[\frac{P(Y=1|X)}{1-P(Y=3|X)}\right] = \beta_{01} + \beta_1 X_1 + ... + \beta_p X_p \approx h(x)$$

$$\log\left[\frac{P(Y=2|X)}{1-P(Y=3|X)}\right] = \beta_{02} + \beta_1 X_1 + ... + \beta_p X_p \approx h(x)$$

Regressão Multinomial: Y em escala ordinal

Exemplo: Grau de melhora de pacientes em tratamento de câncer de mama

$$Y = \begin{cases} 1, & \text{se } \textit{Nenhuma} \\ 2, & \text{se } \textit{Alguma} \\ 3, & \text{se } \textit{Acentuada} \end{cases}$$

$$\log\left[\frac{P(Y=1|X)}{1-[P(Y=2|X)+P(Y=3|X)]}\right] \approx h(x)$$

$$\log\left[\frac{P(Y=1|X)+P(Y=2|X)}{1-P(Y=3|X)}\right] \approx h(x)$$

Regressão

- Em problemas em que Y assume valores contínuos em um conjunto C

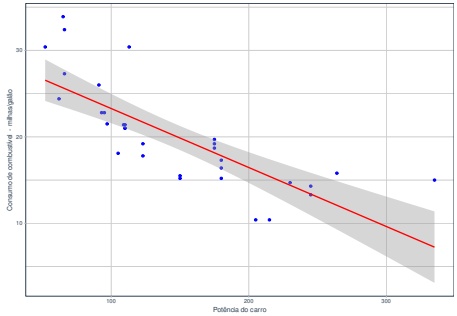
a. Rendimento, em dólares, de um investimento financeiro
b. Nível do biomarcador para câncer de próstata
c. Consumo de combustível de uma amostra de veículos
d. Peso de produtos de uma determinada fábrica
e. Resistência de vigas em uma construção civil

Regressão Linear simples

- Tem a seguinte forma:

$$Y = \beta_0 + \beta_1 X$$

1. β_0 é o intercepto da reta.
2. β_1 é o coeciente angular da reta.



Regressão Linear múltipla

- Tem a seguinte forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

- Em outras palavras:
  1. Considere a problema contínuo onde $Y \in [-\infty, \infty]$.
  2. Para um dado vetor de cararcterísticas x, queremos estimar $E(Y = c \mid x)$.
  3. Output do modelo será $\hat{Y}$.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p \approx h(x)$$

Árvore de Decisão

- Cria conjunto de regras que envolvem a **estratificação** ou **segmentação** do espaço de predição em regiões simples;
- Pode ser aplicado a problemas de classificação
- Pode ser aplicado a problemas de regressão

Árvore de Decisão

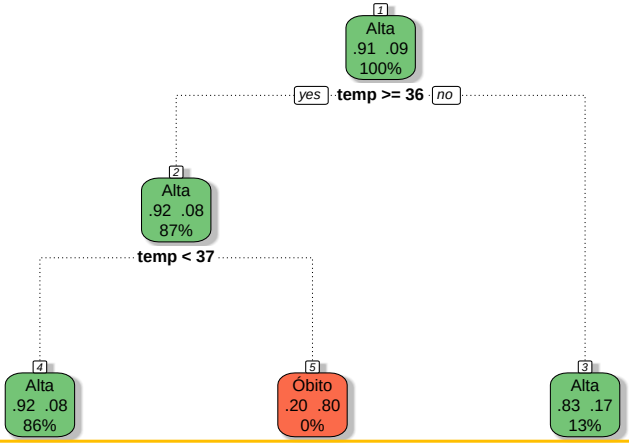
- Cria conjunto de regras que envolvem a **estratificação** ou **segmentação** do espaço de predição em regiões simples;
- Pode ser aplicado a problemas de classificação
- Pode ser aplicado a problemas de regressão

Árvore de Decisão

- Cria conjunto de regras que envolvem a **estratificação** ou **segmentação** do espaço de predição em regiões simples;
- Pode ser aplicado a problemas de classificação
- Pode ser aplicado a problemas de regressão

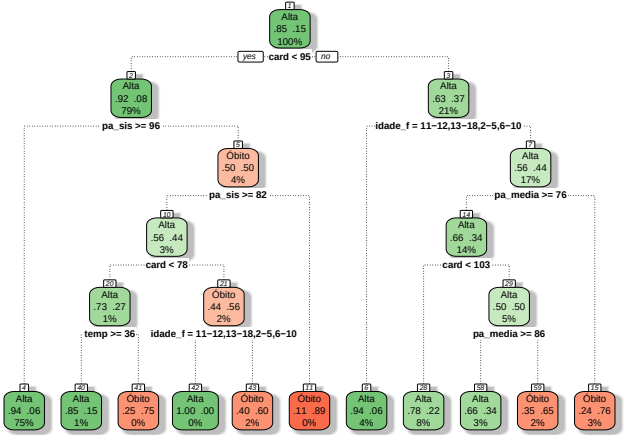
Árvore de Decisão

- Y - Alta(0), Óbito(1)
- X - Temperatura do paciente em graus celsius



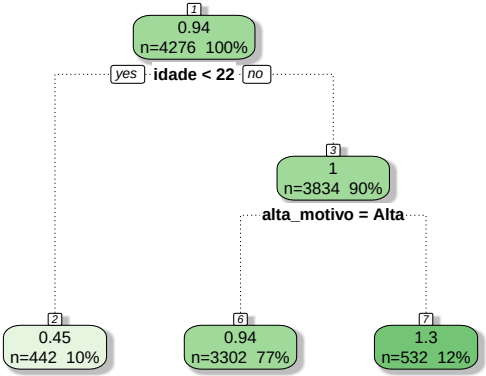
Árvore de Decisão

- Y - Alta(0), Óbito(1)
- X - Vetor de múltiplas características



Árvore de Decisão

- Y - Resultado do exame Creatinina - escala contínua
- X - Vetor de múltiplas características



Árvore de Decisão: Classificação

Medidas de impureza

- Ganho de Informação

$$Ganho(S, X) = Entropia(S) - \sum_{v \in Valores(x)} \frac{|\ S_v\ |}{|\ S\ |} Entropia(S_v)$$

em que $Entropia = \sum_{i=1}^{c} p_i \log_2 p_i$, onde p_i é a probabilidade da classe C_i pertencer a um conjunto S.

Árvore de Decisão: Classificação

Medidas de impureza

- Critério de Gini

$$G = \sum_{c=1}^{C} \hat{p}_{mc} \left(1 - \hat{p}_{mc}\right),$$

- Mede a variância total nas C classes.
- Esse índice apresenta menor valor quando os $\hat{p}_{mc}$'s são próximos de 0 ou 1.
- Valores pequenos indicam que o nó contém predominantemente observações de uma classe.

Árvore de Decisão: Regressão

- O objetivo é encontrar partições $C_1, \ldots, C_J$ que minimizam a:

$$SQRes = \sum_{i:x_i \in S_1(j,s)} (y_i - \hat{y}_{C_1})^2 + \sum_{i:x_i \in S_1(j,s)} (y_i - \hat{y}_{C_2})^2,$$

em que $C_1(j,s) = X \mid X_j < s$ e $C_2(j,s) = X \mid X_j \geq s$

Árvore de decisão

Diversos algoritmos

- ID3 (Quinlan 1979)
- CART – Classification and Regression Trees (Breiman et al. 1984)
- C4.5 (Quinlan 1993)
- CHAID (Chi Square Automatic Interaction Detection)

Árovre de decisão: algumas considerações

1. Podem ser aplicadas em problemas de regressão e classificação.
2. Lidam bem com dados faltantes.
3. São simples e úteis para interpretação.
4. As predições não são competitivas com outros algoritmos.
5. Serve de base para outros métodos, como:
   - *Bagging*
   - *Random Forests*
   - *Boosting*

Bagging

A ideia do *bagging* baseia-se nesse conceito, aumentamos a precisão final ao utilizar a média das predições (várias árvores, neste caso). Fazemos isso gerando repetidas amostras através do Bootstrap. O processo completo é descrito em seguida

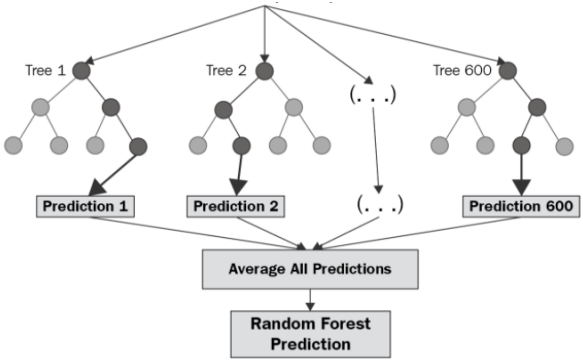
- Algoritmo
  1. Gerar B conjuntos de observações (bootstrapped).
  2. Treinar o modelo, a fim de obter a predição no ponto x;
  3. Calcular a média das predições, que chamamos de *bagging*:

$$\hat{h}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{h}^b(x),$$

em que $\hat{h}^b(x)$ é o modelo de árvore, construído com a b-ésima amostra *bootstrap*.

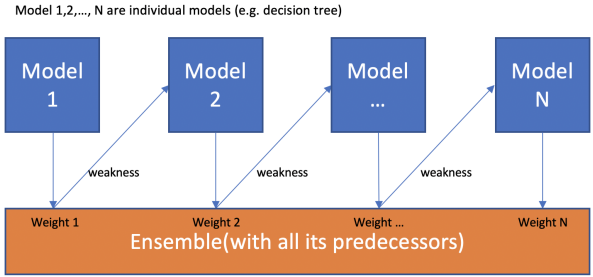
Random Forests

- No ***Random forests***, para cada partição, temos uma seleção aleatória de m preditores, de um total de p (tipicamente, $m \approx \sqrt{p}$);
- Assim, forçamos com que diferentes preditores sejam escolhidos. Se $m = p$, estaremos no método *Bagging*



Boosting

- A informação da árvore anterior é utilizada para construção da próxima.
- Propor um modelo básico *(weak learner)* e o aprimorar em cada iteração.
- Consiste em filtrar os resultados corretos, e concentrar-se naqueles que o modelo não soube lidar.
- Ou seja, desenvolver os pontos fracos do algoritmo.



Boosting

- Cada etapa propor um novo modelo (somando-o ao antigo), baseando-se nos resíduos.
-

$$Y = h(x) + \text{resíduo} \tag{1}$$

$$\text{resíduo} = g(x) + \text{resíduo2} \tag{2}$$

- Combinando (1) e (2)

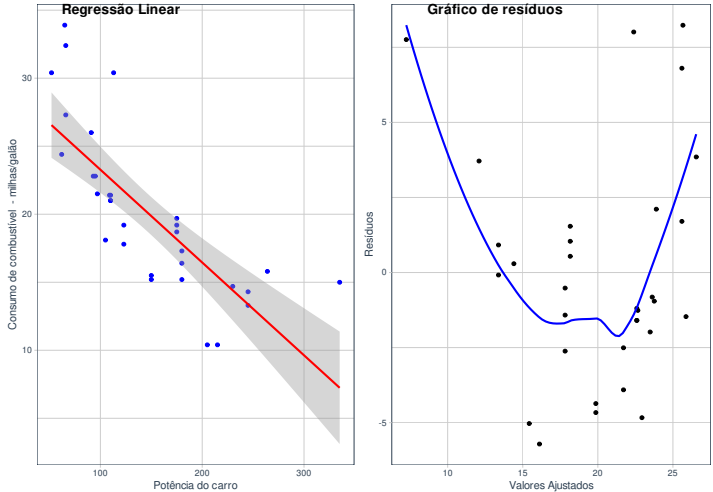
$$Y = h(x) + g(x) + \text{resíduo2}$$

- $h(x)$ foi atualizada com uma parte do resíduo, ou seja

$$h(x)^{(2)} = h(x)^{(1)} + g(x)$$

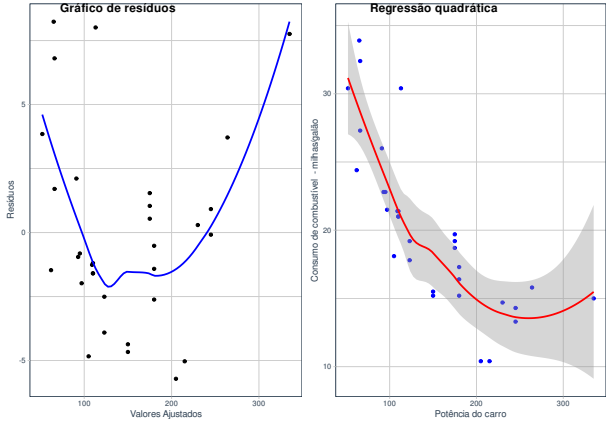
Gradiente Boosting Machine

- $mpg = \beta_0 + \beta_1(potencia) + Residuo$



Gradiente Boosting Machine

- $Residuo = \beta_2(potencia)^2 + Residuo2$



- $mpg = \beta_0 + \beta_1(potencia) + \beta_2(potencia)^2 + Residuo2$

Gradiente Boosting Machine: relação com o **Gradiente**

- Queremos minimizar

$$J(y_i, h(\mathbf{x})) = \frac{1}{2n}\sum_{i=1}^{n}[y_i - h(\mathbf{x}_i)]^2$$

- Derivando com relação a $h(\mathbf{x}_i)$

$$\frac{\partial J(\mathbf{y}, h(\mathbf{x}))}{\partial h(\mathbf{x}_i)} = h(\mathbf{x}_i) - y_i$$

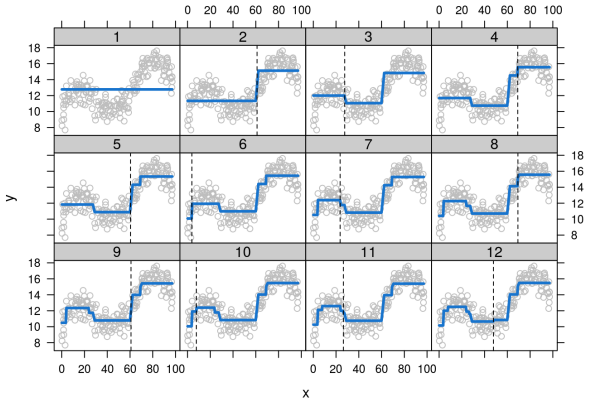
- Podemos interpretar os resíduos como o negativo do gradiente

$$\begin{aligned} residuos &= y_i - h(\mathbf{x}_i) \\ &= -\frac{\partial J(y_i, h(\mathbf{x}))}{\partial h(\mathbf{x}_i)} \end{aligned}$$

Gradiente Boosting Machine: Em resumo

$$\text{resíduo} \quad \Leftrightarrow \quad \text{negativo do gradiente}$$

$$\text{Atualizar} \quad h(\mathbf{x}_i) \quad \text{com o resíduo} \quad \Leftrightarrow \quad \text{Atualizar} \quad h(\mathbf{x}_i) \quad \text{com o negativo do gradiente}$$



Desafio 4

Com os dados coletados no desafio 1, faça:

- Uma análise utilizando o algoritmo *K-means*
- Um modelo de aprendizado supervisionado.
- Compare seus resultados utilizando os métodos de árvores.
  1. *Random forests.*
  2. *Gardient Boosting Machine.*
- Se for um problema de classificação, compare também com regressão logística ou multinomial.
- Se for um problema de regressão, compare também com regressão linear.
- Em ambos os casos decida por um dos modelos e explique a sua escolha.

Referências

1. Hastie, T., Tibshirani, R. e Friedman, J., **The Elements of Statistical Learning**, 2009.
2. James, G., Witten, D., Hastie, T. e Tibshirani, **An Introduction to Statistical Learning**, 2013.
3. L. Breiman. Statistical modeling: **The two cultures. Statistical Science**, 16(3):199-231, 2001.
4. Lantz, B., **Machine Learning with R**, Packt Publishing, 2013.
5. Tan, Steinbach, and Kumar, **Introduction to Data Mining**, Addison-Wesley, 2005.
6. Mitchell, T. M. (1997). **Machine Learning**. McGraw-Hill.
7. Wickham, H.; Grolemund, G. **R for data science: import, tidy, transform, visualize, and model data**. " O'Reilly Media, Inc.", 2016.