

Introdução ao aprendizado de máquina



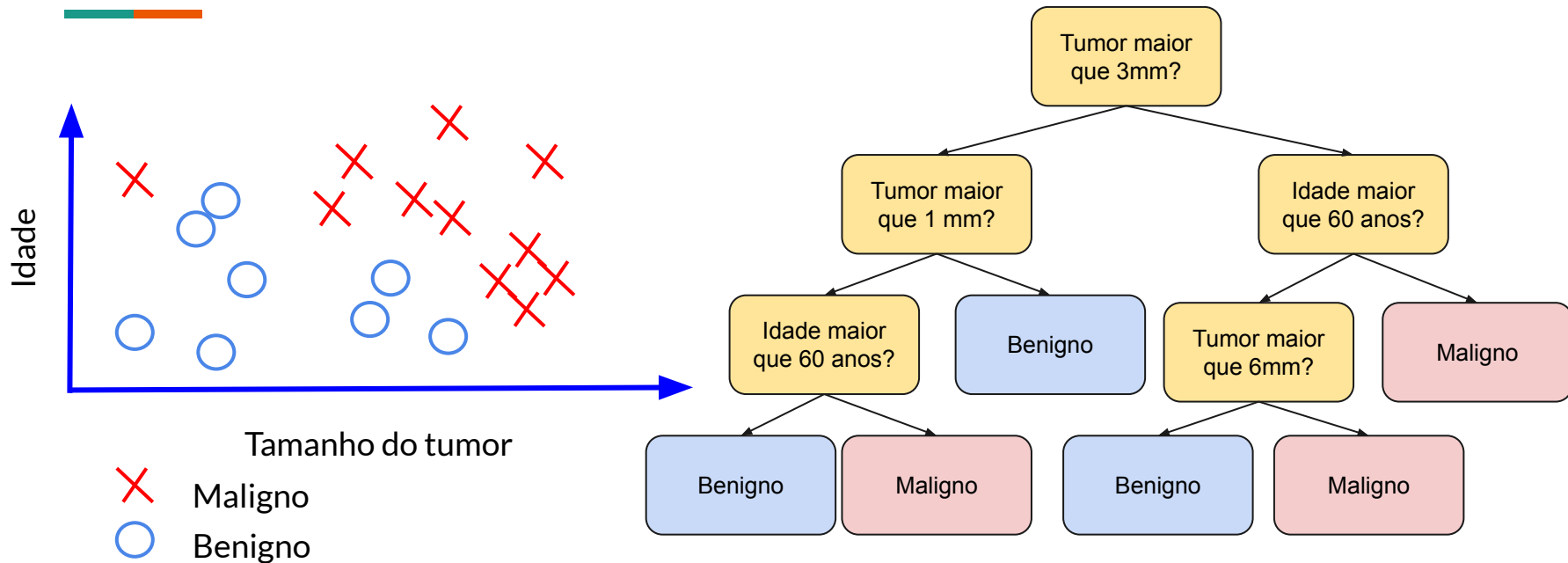
Aula 7

- Árvores de Decisão
- Gini e Entropia
- Conjuntos de modelos (ensembles)
- Florestas Aleatórias
- AdaBoosting
- Gradient Boosting Trees

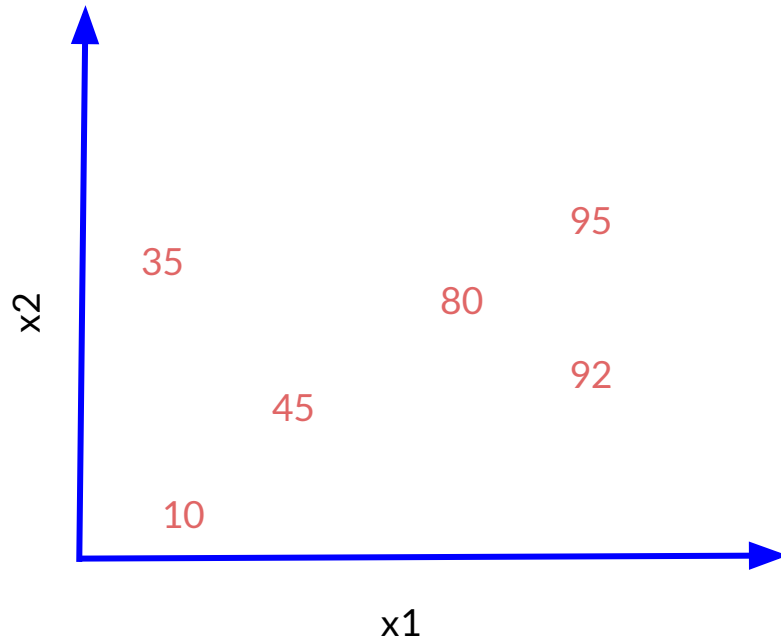


Exemplos

Exemplo de árvore de decisão com classificação



Árvore de decisão com regressão





Algoritmo

Pseudo-código de uma árvore de decisão

ArvoreDecisao(dados):

Determine o erro máximo

Enquanto erro for maior que o erro máximo

 Loop por todas as folhas

 Loop por todas as variáveis

 Loop por todos os critérios de decisão

 Encontre o critério que mais diminui o erro

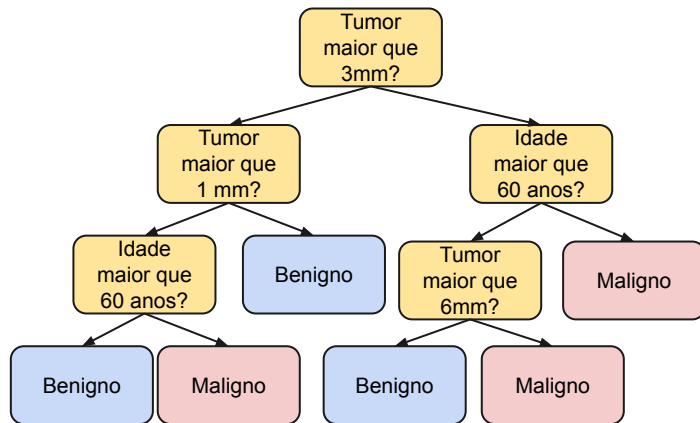
 Divida a folha nesta variável usando esse critério

 Adicione à árvore o par variável-critério

 Determine a previsão das novas folhas como a valor médio de suas observações

 Cheque para ver se o erro é menor que o erro máximo

Retorne a árvore final





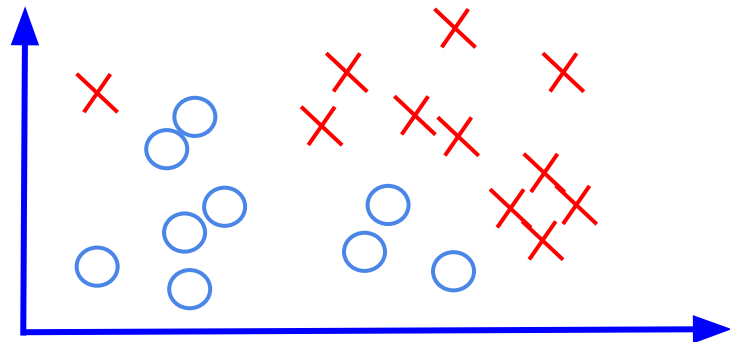
Métricas de decisão

Impuridade Gini



Impuridade Gini

$$\begin{aligned} G(p) &= \sum_{i=1}^J p_i \sum_{i \neq j} p_i \\ &= \sum_{i=1}^J p_i (1 - p_i) \\ &= 1 - \sum_{i=1}^J p_i^2 \end{aligned}$$



Ganho de informação



Ganho de informação

Ganho de Informação = $H(\text{pre divisão}) - \text{media ponderada } H(\text{pos divisão})$

$$H(p) = - \sum_{i=1}^J p_i \ln_2(p_i)$$

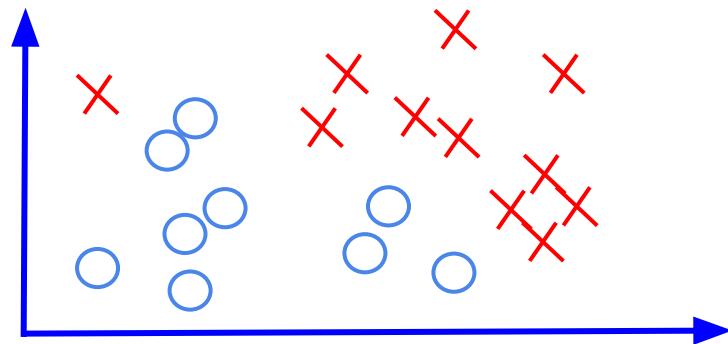
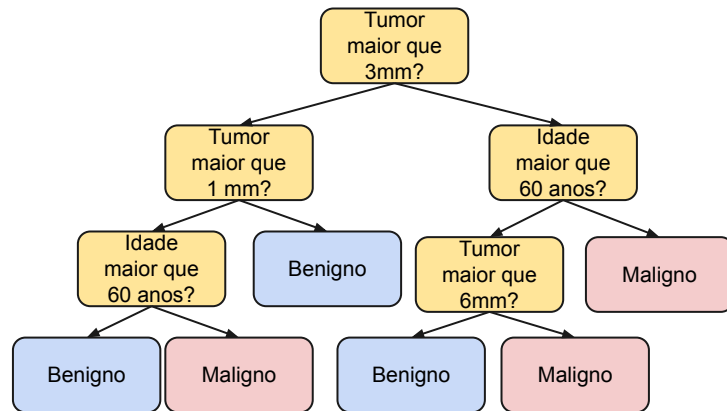


Regularização

Regularização em árvores de decisão

Técnicas de regularização:

- Profundidade máxima da árvore
- Entropia ou Gini mínimo para divisão de uma folha
- Mínimo número de elementos em uma folha
- Máximo número de características analisadas para uma divisão
- Pesos nas classes, possivelmente “balanceadas”





Prós e Contras

Prós e contras de árvores de decisão



Prós:

- Fácil de interpretar
- Poderosas - excelentes resultados no conjunto de treino
- Não gera previsões absurdas
- Versátil

Cons

- Sobreajuste acontece com frequência
- Sensível a decisão de como separar conjunto de treino e validação
- Instável
- Assume ortogonalidade

Instabilidade e ortogonalidade

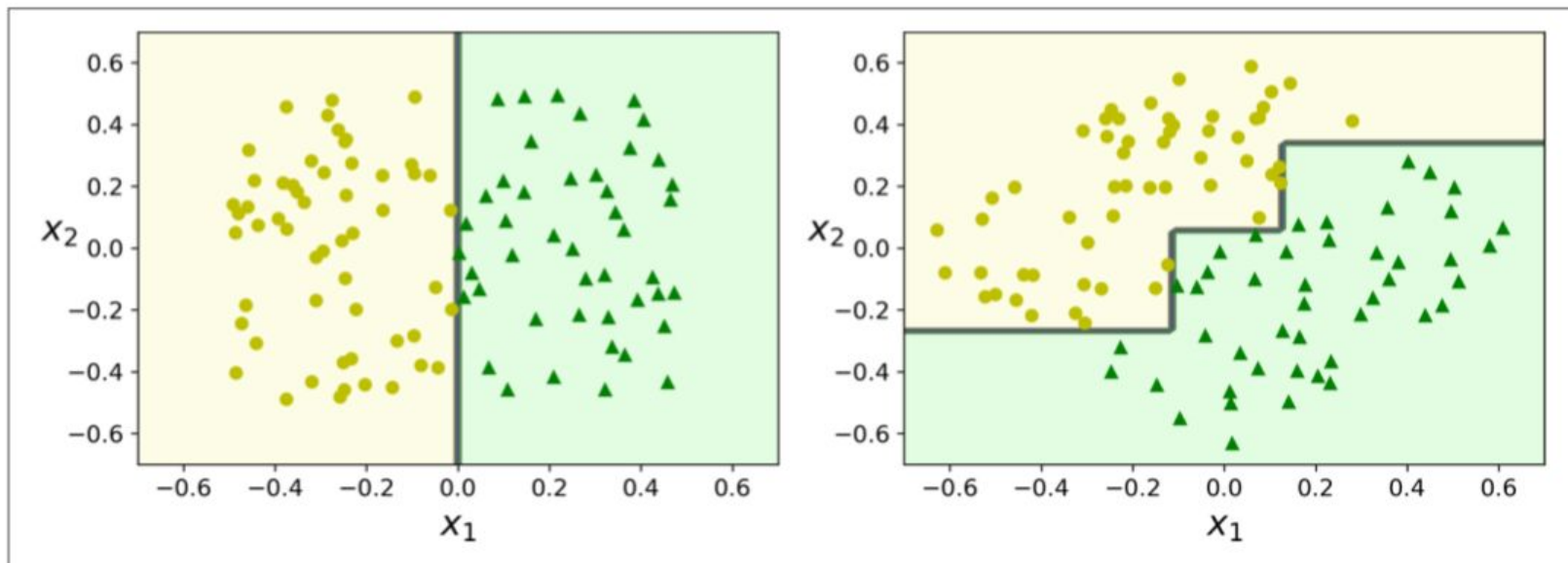
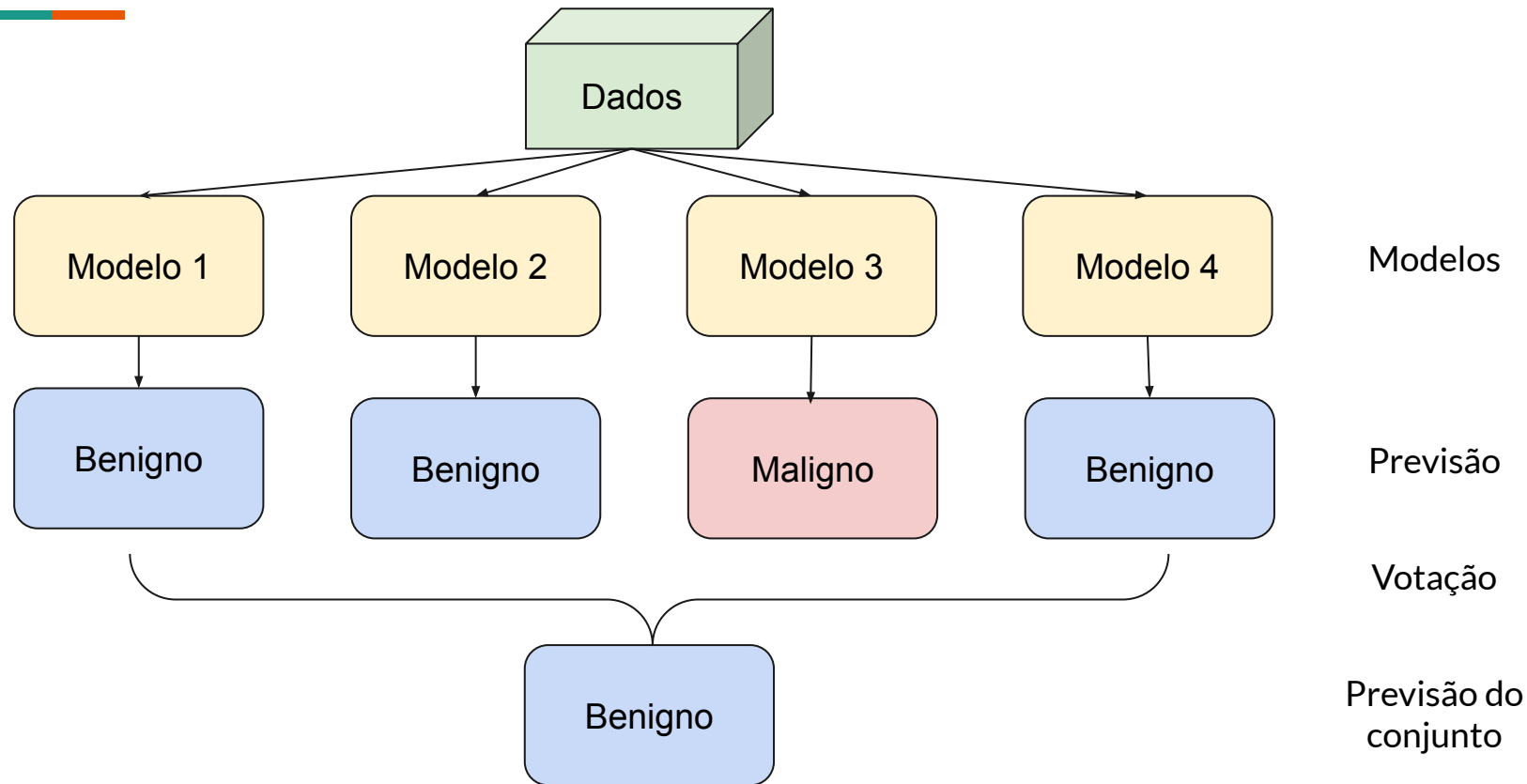


Figure 6-7. Sensitivity to training set rotation



Conjuntos de modelos (Ensembles)

Vizualização de conjuntos de modelos



Conjuntos de modelos

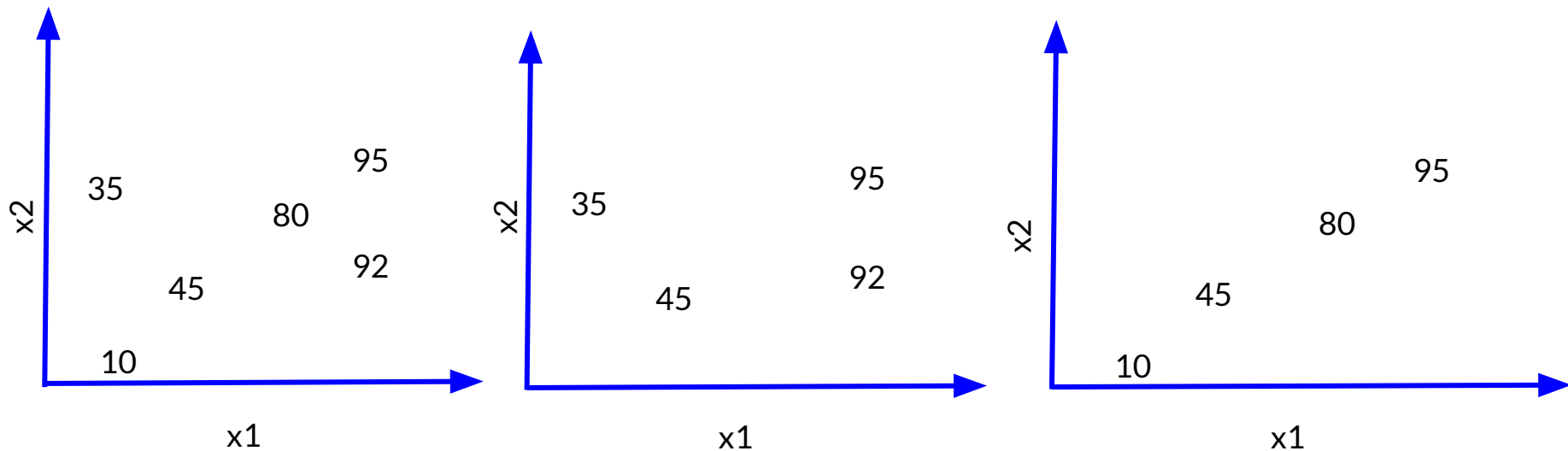


- Agrupar modelos diferentes pode melhorar a eficiência
- Frequentemente o conjunto de modelos tem uma acurácia melhor que o melhor modelo individual
- Vários conjuntos de modelos vencem competições Kaggle
- Podemos tirar a média das previsões (hard voting) ou a média das probabilidades (soft voting)
- Modelos podem ter o mesmo peso ou podemos estimar a importância de cada modelo
- Quão melhor a métrica dos modelos, melhor a métrica do conjunto
- Quão menos correlacionados os modelos, melhor a métrica do conjunto



Encasamento (bagging) e colagem (pasting)

Exemplo de amostragem



Bagging e Pasting



- Bagging ocorre quando criamos um conjunto de modelos gerados através do mesmo algoritmo, mas com diferentes amostras com repetição da base de dados
- Pasting é similar, mas a subamostra de cada modelo é feita sem repetição
- As observações não selecionadas em cada modelo podem servir como mais um conjunto para testar a performance, métrica out-of-bag

Fronteira de decisão com bagging

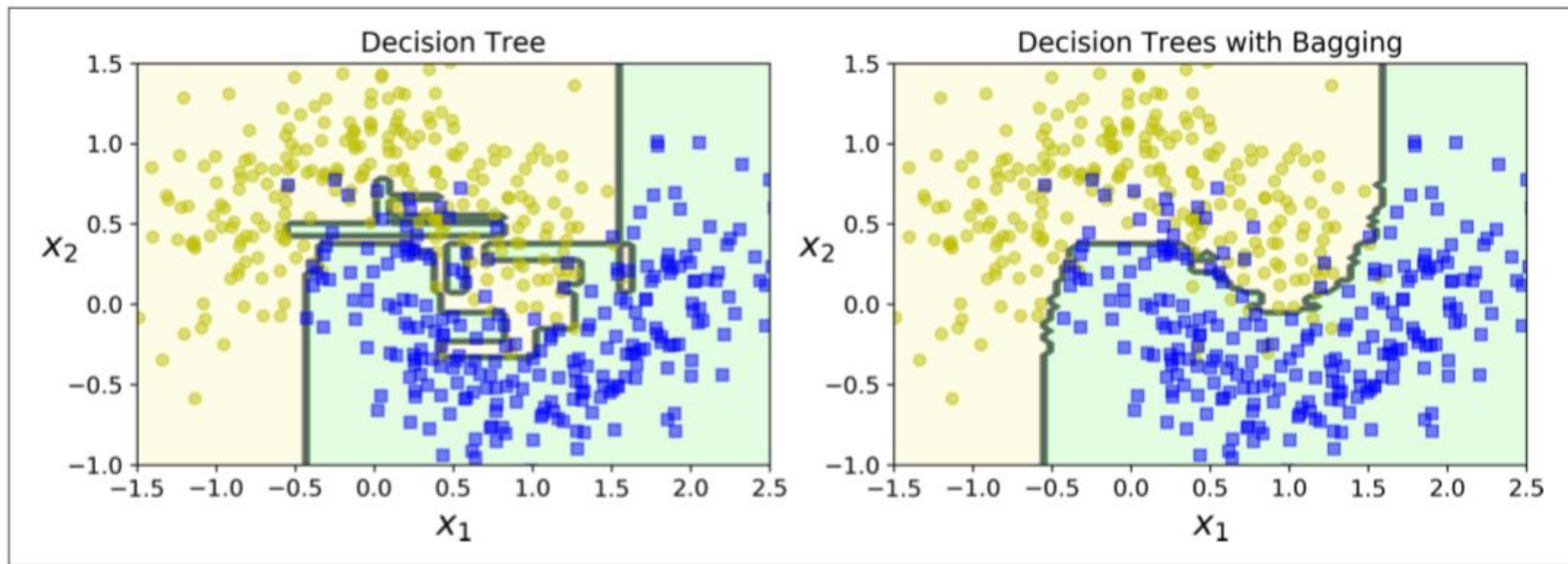


Figure 7-5. A single Decision Tree versus a bagging ensemble of 500 trees

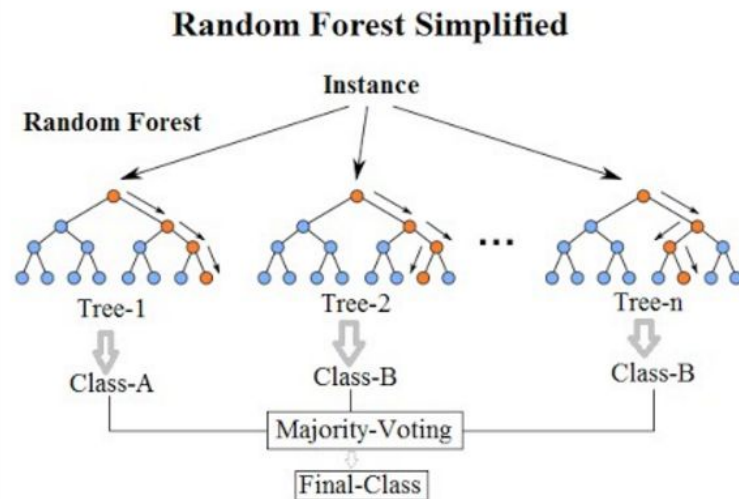


Florestas Aleatórias

Florestas aleatórias

Floresta aleatória é um bagging de árvores de decisão

- Escolhemos o número de árvores a serem treinadas (ex: 40 árvores)
- Para cada árvore selecionamos uma amostra aleatória com repetição do tamanho da base de dados inicial
- O número de características em cada árvore também pode ser selecionado aleatoriamente.
- Cada observação passa por cada árvore e recebe uma previsão.
- No final tiramos a média das previsões e essa é a previsão da floresta aleatória



Árvores extremamente aleatórias



- Um modelo ainda mais aleatório é chamado de árvores extremamente aleatórias (Extremely randomized trees)
- Não apenas as características são selecionadas aleatoriamente, mas também os critérios de cada característica para considerar a divisão da folha
- Isso leva a um modelo com ainda mais vies e menos variância, mas quando tiramos a média de vários modelos assim, os resultados são promissores



Método de reforço (Boosting Methods)

Métodos de reforço (boosting)



- Boostings são conjuntos de modelos treinados sequencialmente
- Eles melhoram a performance do modelo anterior
- Consequentemente, eles não podem rodar em paralelo
- Boosting melhoram significativamente a performance de árvores de decisão, mas as vezes levam a overfitting

Adaboosting

- Adaboosting começa treinando uma árvore de decisão
- Em seguida, em cada iteração, retiramos uma amostra aleatória onde cada observação tem um probabilidade diferente de ser escolhida
- Observações onde o algoritmo produz um resultado ruim têm um peso maior
- No final, computamos a média ponderada entre todas as árvores

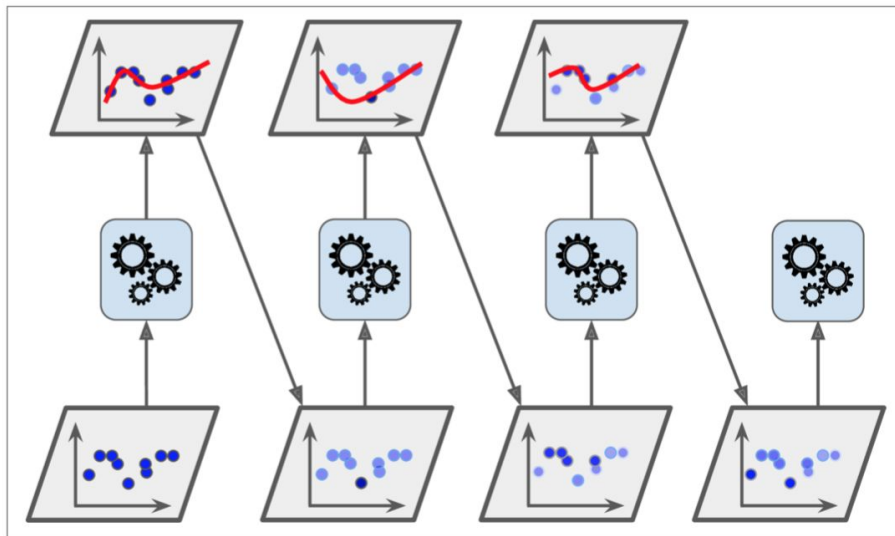


Figure 7-7. AdaBoost sequential training with instance weight updates

Gradient Boosting

- Gradient Boosting também começa com uma árvore de decisão
- Em cada iteração, treinamos uma nova árvore tentando prever os **erros** da árvore anterior
- No final, somamos as previsões das árvores
- Gradient Boosting ganhou várias competições de machine learning (Kaggle)
- Uma implementação otimizada de gradient boosting está disponível no pacote XGboost em python

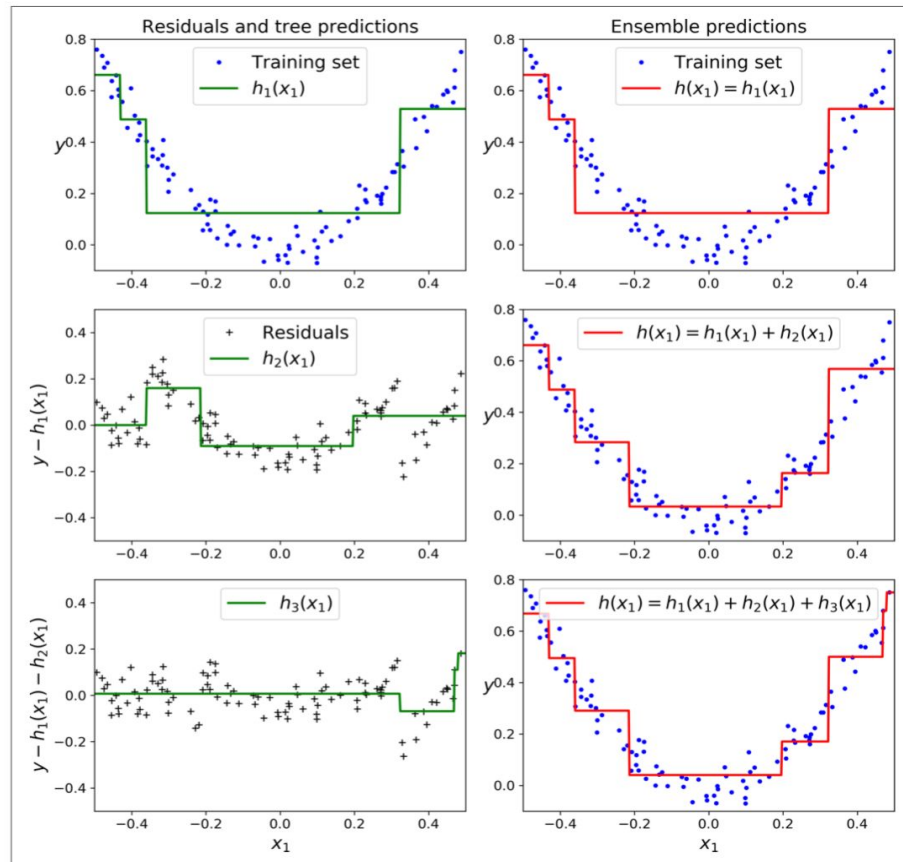


Figure 7-9. Gradient Boosting