

Árvores de Decisão e Métodos de Ensemble

Nicolas Moreira Branco

EEL7513/EEL7514 - Tópico Avançado em Processamento de Sinais

EEL410250 - Aprendizado de Máquina

EEL / CTC / UFSC

Árvores de decisão

- Conceitos gerais
- Exemplo de aplicação
- Probabilidade de classe
- Algoritmo CART
- Complexidade computacional
- Impureza Gini vs entropia
- Regularização
- Regressão
- Instabilidade

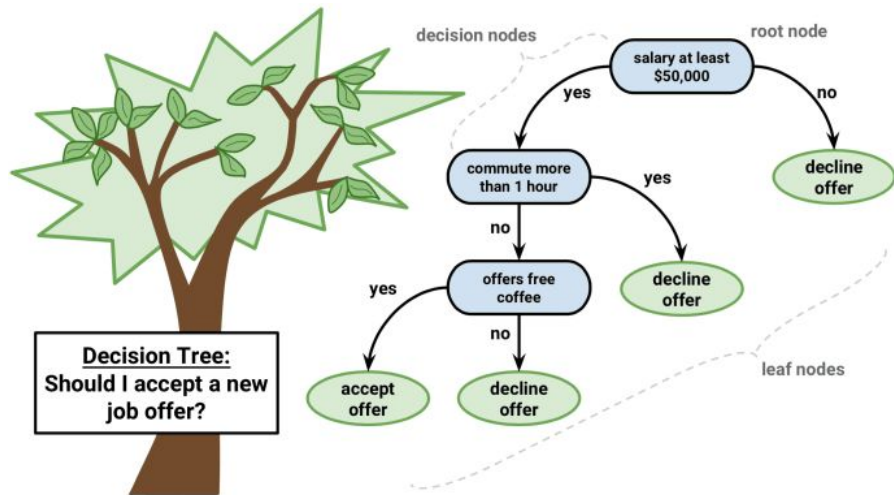
Ensemble learning

- Conceitos gerais
- Poder dos weak learners
- Votação de classificadores
- Bagging and pasting
- Amostragem de atributos
- Random forest
- Extremely randomized trees
- Importância de cada atributo
- Boosting
- Adaboost
- Gradient Boosting
- Stacking

Árvores de decisão

Conceitos gerais

- Raiz
 - Início da árvore
- Nós de decisão
 - Usualmente binárias, mas podem ter 3 filhos em alguns algoritmos
- Folhas
 - Resultado (classe ou regressão)
- Modelo *White Box*
- Não precisa de normalização dos atributos



Exemplo de aplicação

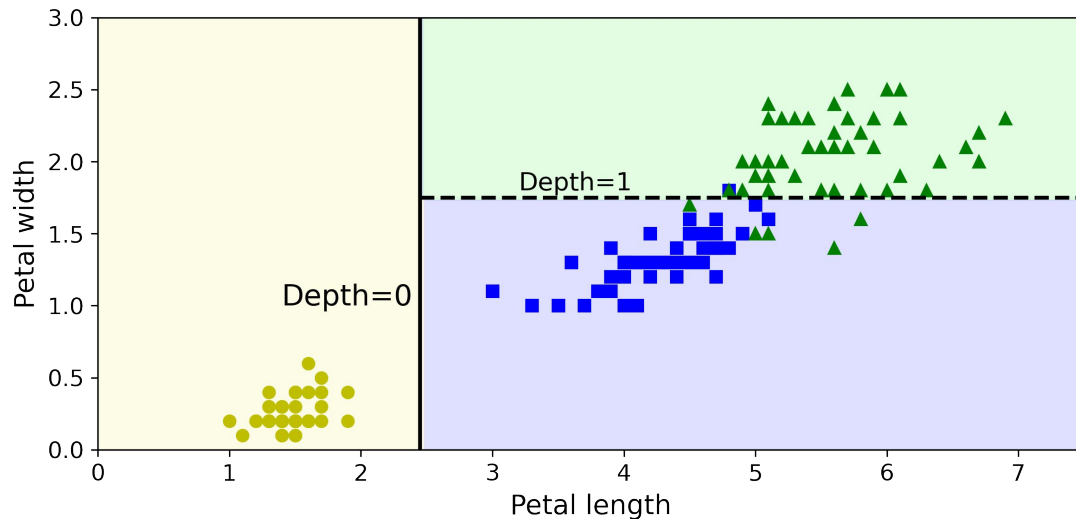
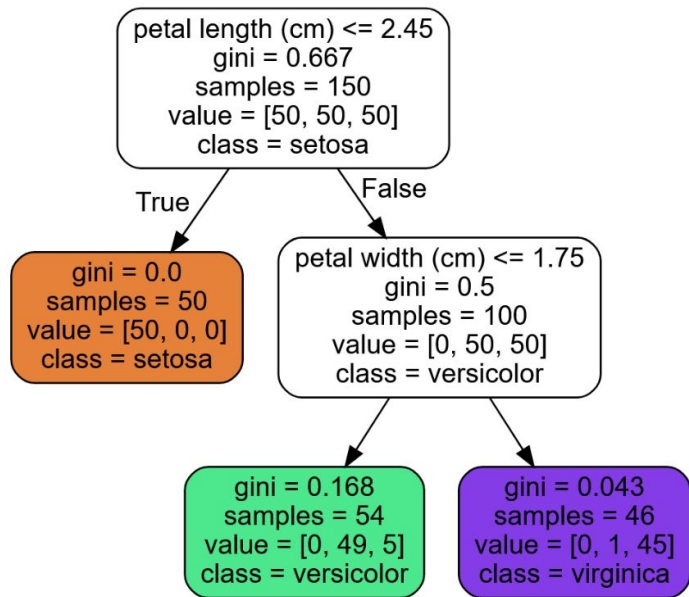
Dataset IRIS

- Tipos de plantas
- 4 atributos
- 3 classes



Exemplo de aplicação

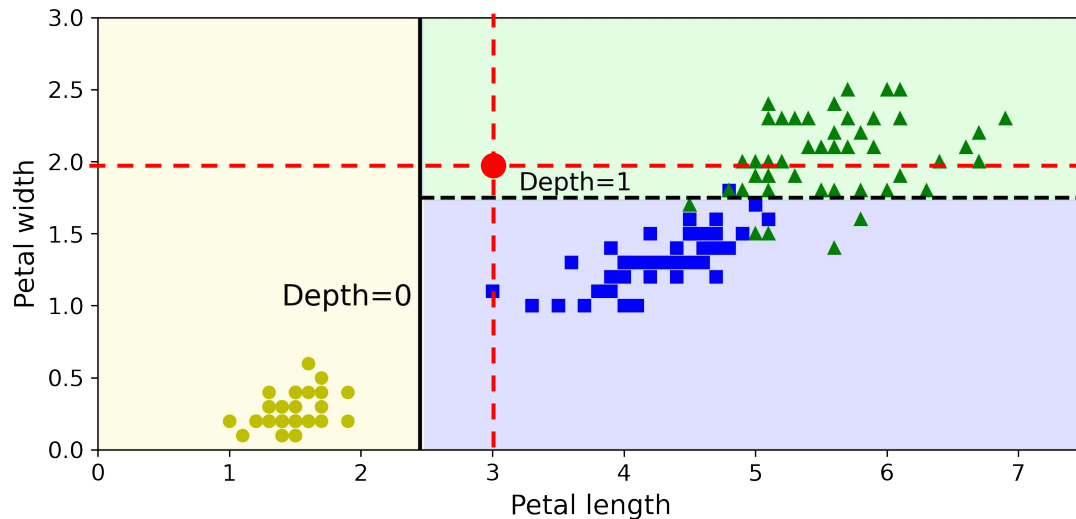
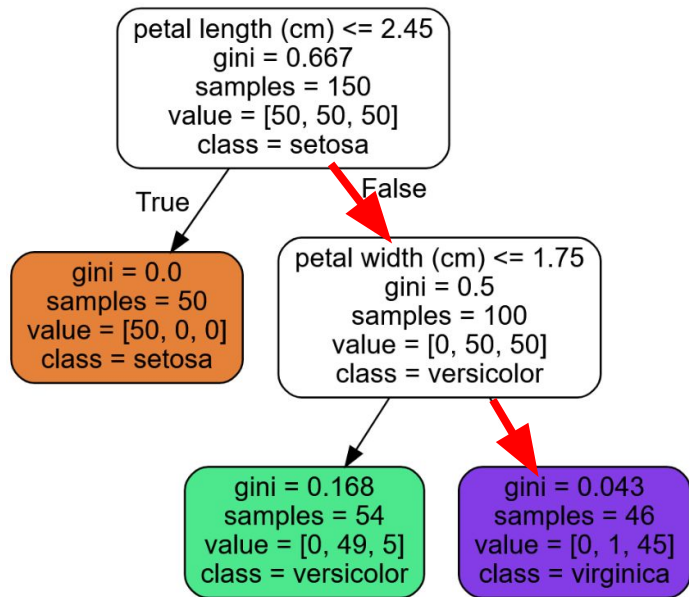
- Árvore de decisão com 2 níveis de profundidade



Exemplo de aplicação


Petal length = 3

Petal width = 2



Probabilidade de classe

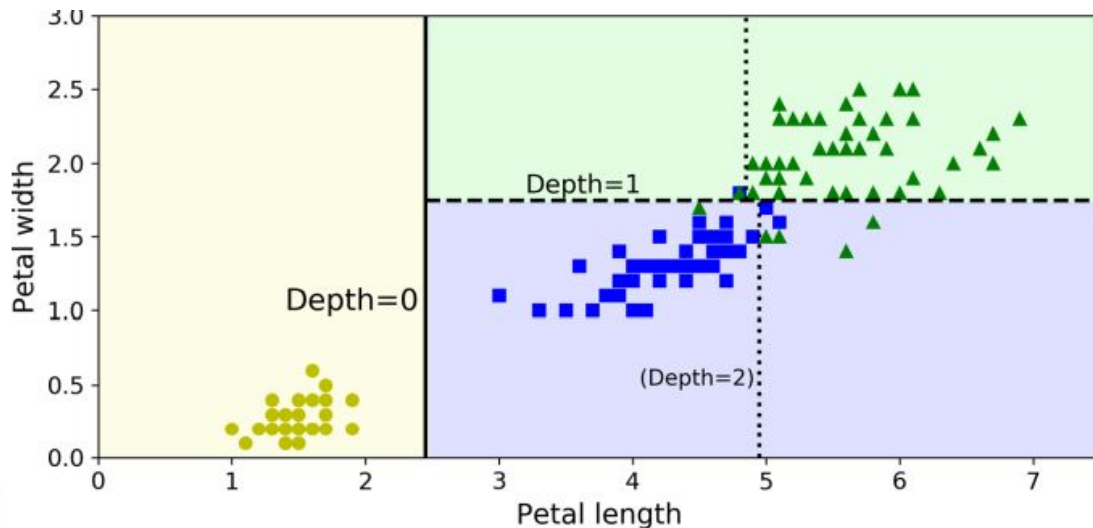
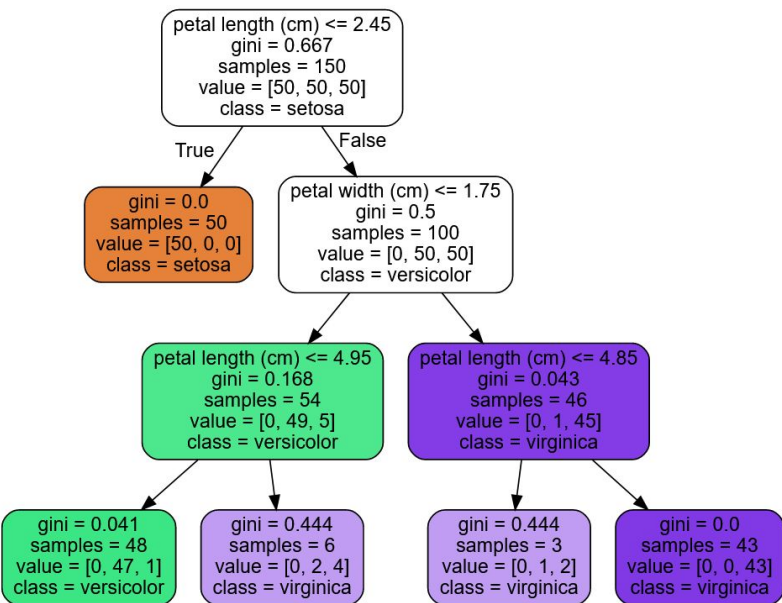
- Acha a folha referente a entrada
- Estima com base nos dados de treinamento
 - Iris Setosa
 - 0/46 - 0%
 - Iris Versicolor
 - 1/46 - 2.2%
 - **Iris virginica**
 - 45/46 - 97.8%



gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica

Exemplo de aplicação

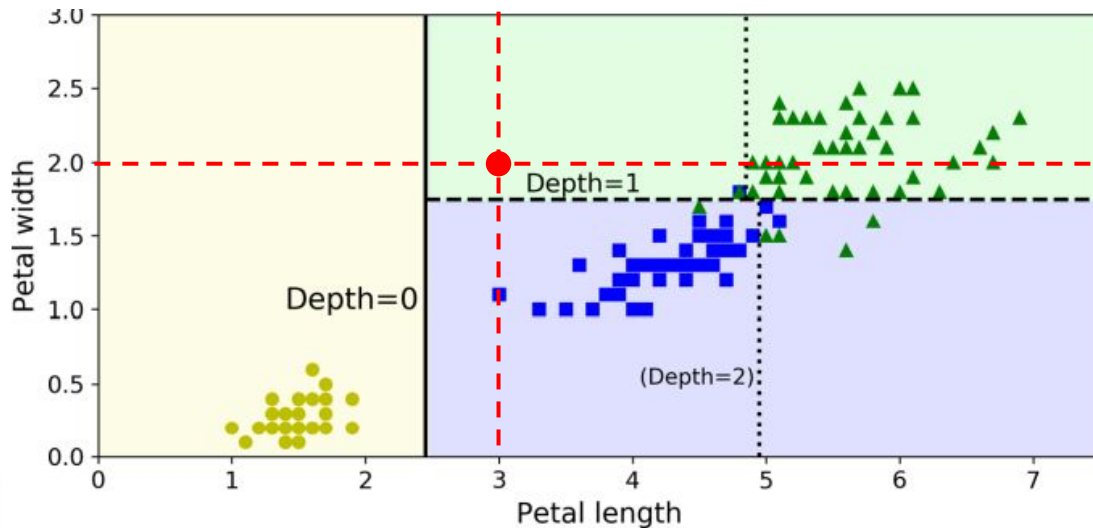
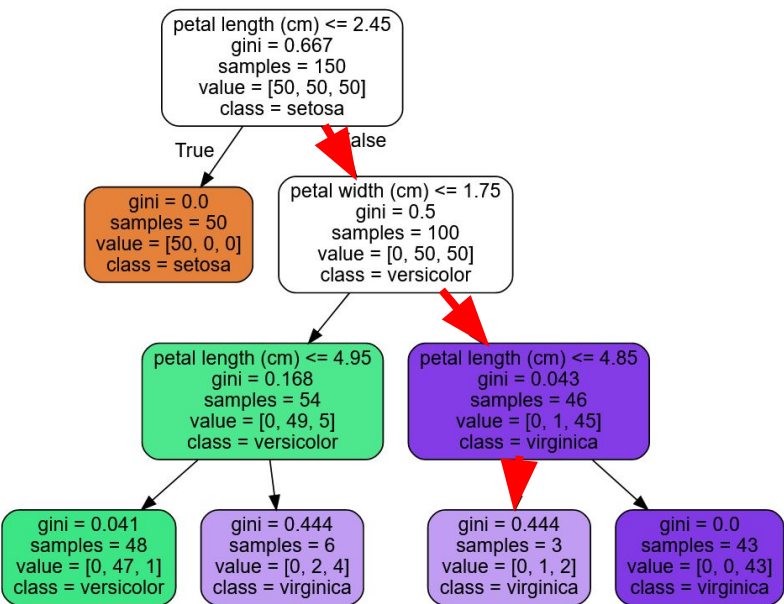
- Árvore de decisão com 3 níveis de profundidade



Exemplo de aplicação

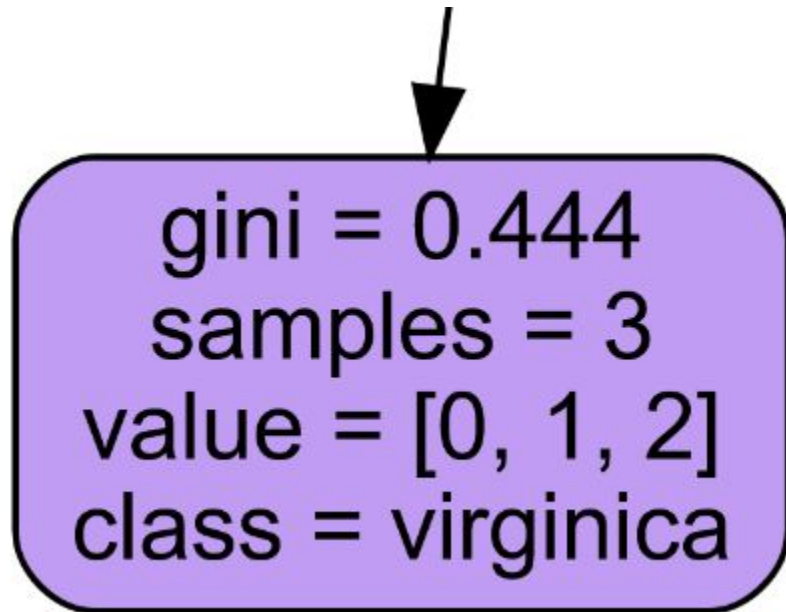
Petal length = 3

Petal width = 2



Probabilidade de classe

- Acha a folha referente a entrada
- Estima com base nos dados de treinamento
 - Iris Setosa
 - 0/54 - 0%
 - Iris Versicolor
 - 1/3 - 33.3%
 - **Iris virginica**
 - 2/3 - 66.6%



Algoritmo CART (Classification and Regression Tree)

- Algoritmo greedy (melhor resultado por level, possivelmente não no geral)
- Divide o dataset usando o atributo k e pelo limiar t_k , procurando o conjunto (k, t_k) que produz os subsets com menor impureza
- Quando termina em um nível repete-se o processo até o último nível

Equation 6-2. CART cost function for classification

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

where $\begin{cases} G_{\text{left/right}} & \text{measures the impurity of the left/right subset,} \\ m_{\text{left/right}} & \text{is the number of instances in the left/right subset.} \end{cases}$

Complexidade computacional

- Predições

- A árvore é atravessada da raiz até as folhas
- Normalmente balanceadas
- $O(\log_2(m))$ para atravessar a árvore
- Como somente uma feature é verificada por nó, a complexidade do modelo é a mesma de atravessar a árvore e é rápido mesmo para uma grande quantidade de atributos

- Treinamento

- Sem limitações nos hiperparâmetros, todas as features são testadas em todos os nós
- Complexidade $O(n \times m \log_2(m))$

Impureza de Gini versus entropia

- Maioria dos casos é similar
- Gini é mais rápido para calcular (bom como padrão)
- Quando diferem, Gini tende a isolar as classes mais frequentes em uma parte da árvore enquanto entropia mantém mais equilibrado

Equation 6-1. Gini impurity

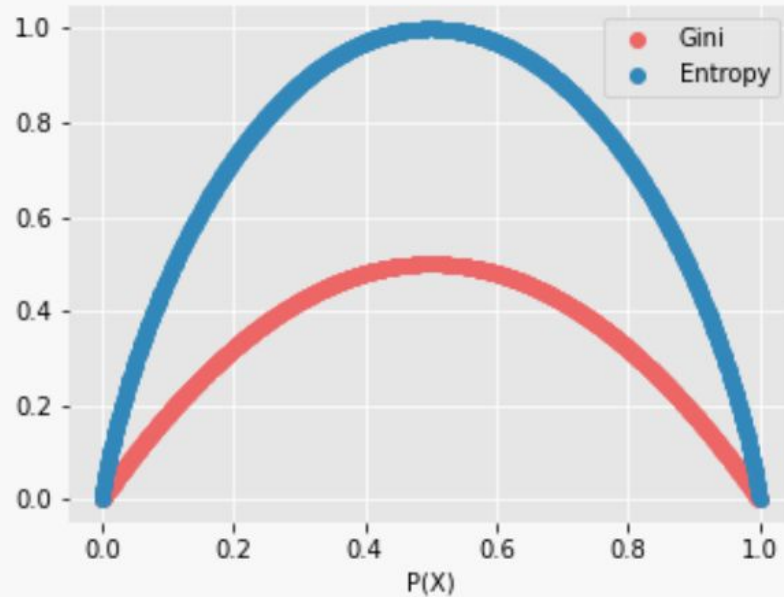
$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

Equation 6-3. Entropy

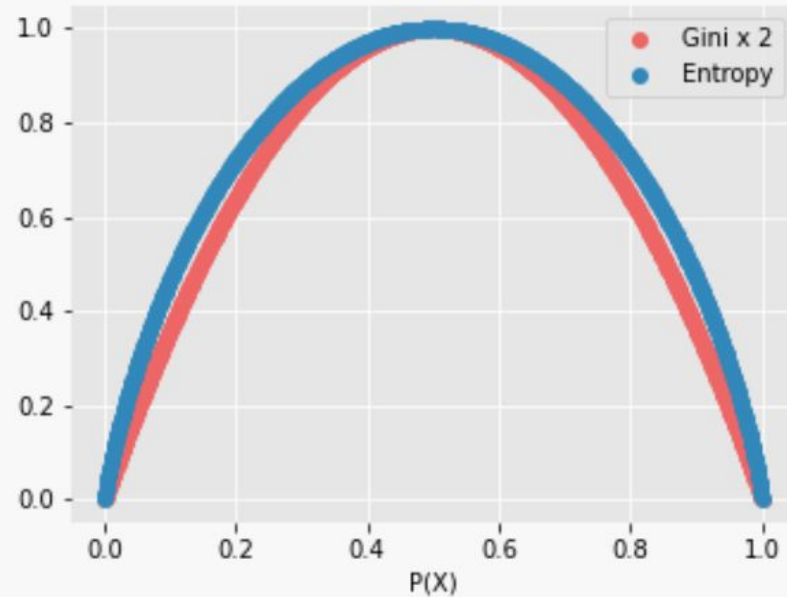
$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^n p_{i,k} \log_2 (p_{i,k})$$

Impureza de Gini versus entropia

Gini [0,0.5]



Entropy [0,1]



Regularização

- Modelo não paramétrico
 - Pode aderir muito bem aos dados de treinamento
 - Sem parâmetros de regularização facilmente pode chegar ao overfit
- Evitar overfitting no treinamento (hiperparâmetros)
 - Profundidade máxima da árvore
 - Mínimo de amostras antes de dividir o nó
 - Mínimo número de amostras que uma folha deve ter
 - Máximo número de folhas
- Também é possível treinar sem limitantes e realizar um pruning posterior

Regularização

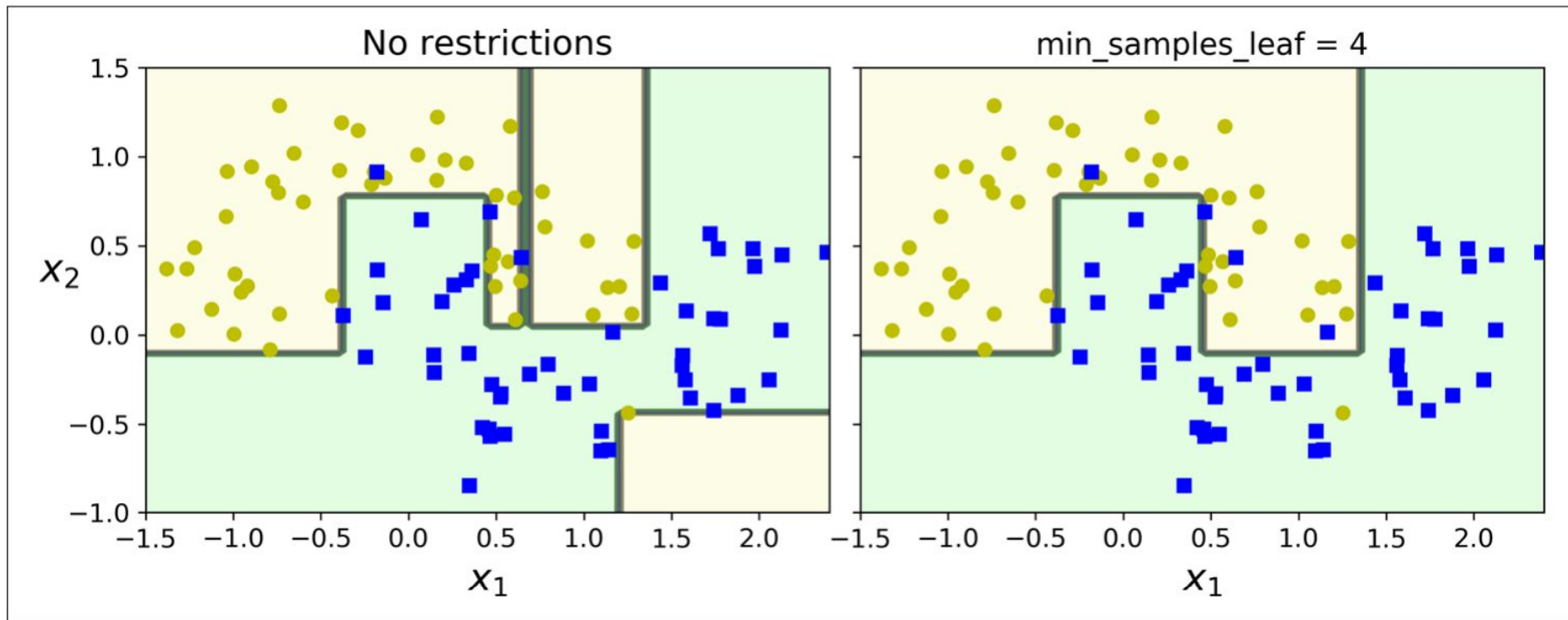


Figure 6-3. Regularization using `min_samples_leaf`

Regressão

- Similar ao de classificação
- Média dos valores do nó
- Usa MSE como impureza

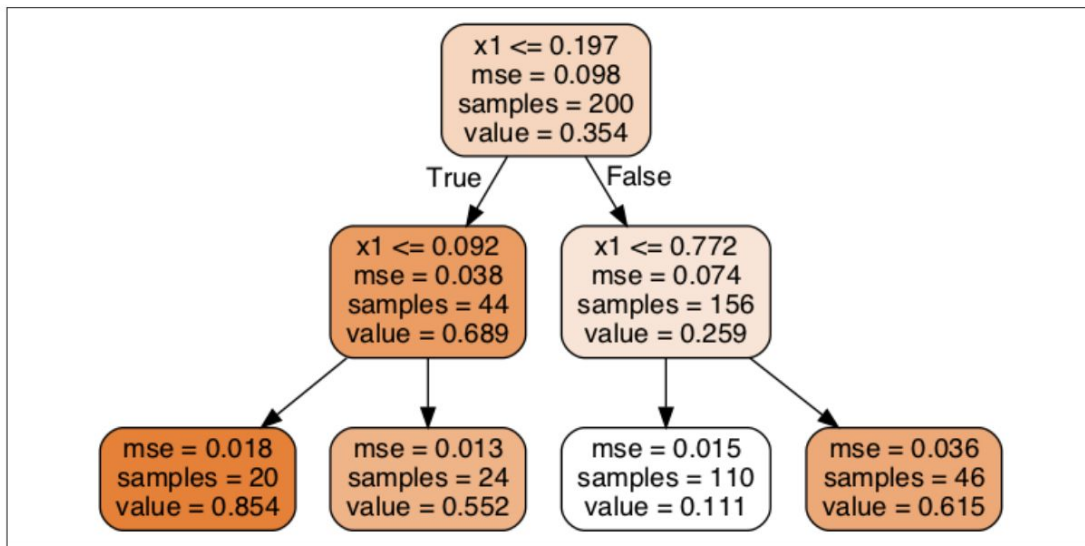


Figure 6-4. A Decision Tree for regression

Regressão

Comparando a profundidade máxima da árvore

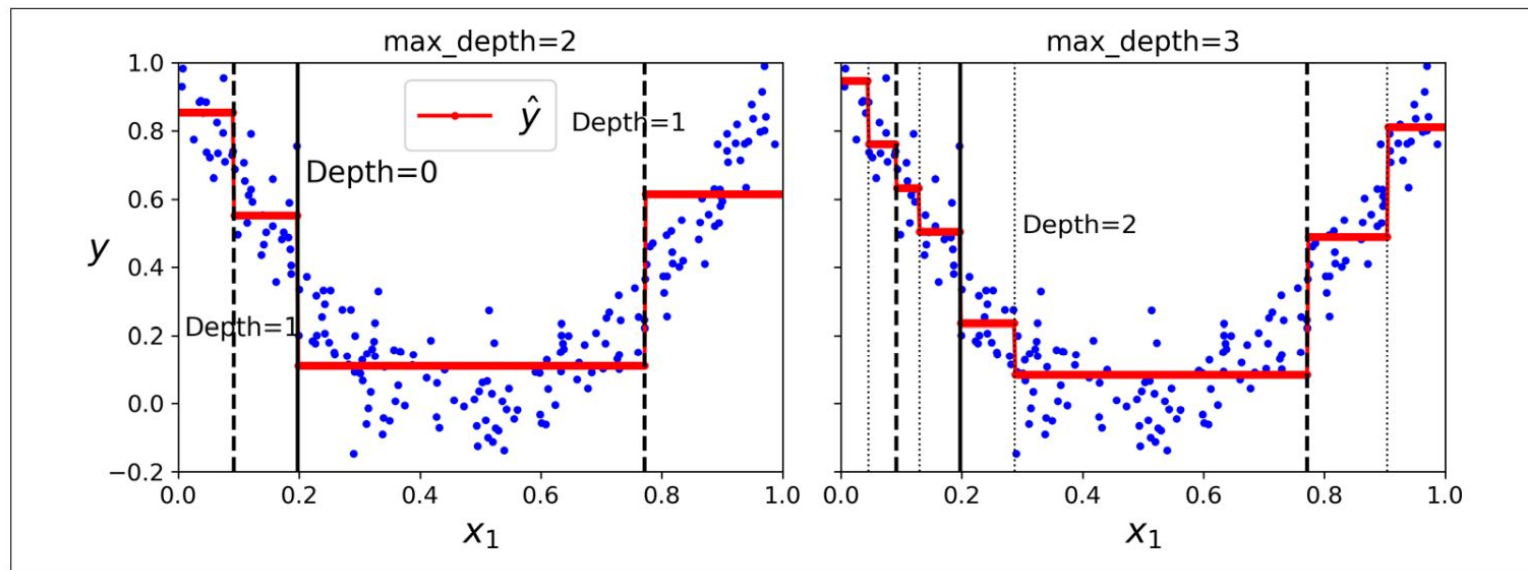


Figure 6-5. Predictions of two Decision Tree regression models

Regressão

CART para regressão, utiliza MSE no local da impureza

Equation 6-4. CART cost function for regression

$$J(k, t_k) = \frac{m_{\text{left}}}{m} \text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m} \text{MSE}_{\text{right}} \quad \text{where} \quad \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

Regressão

Overfitting e hiperparâmetros

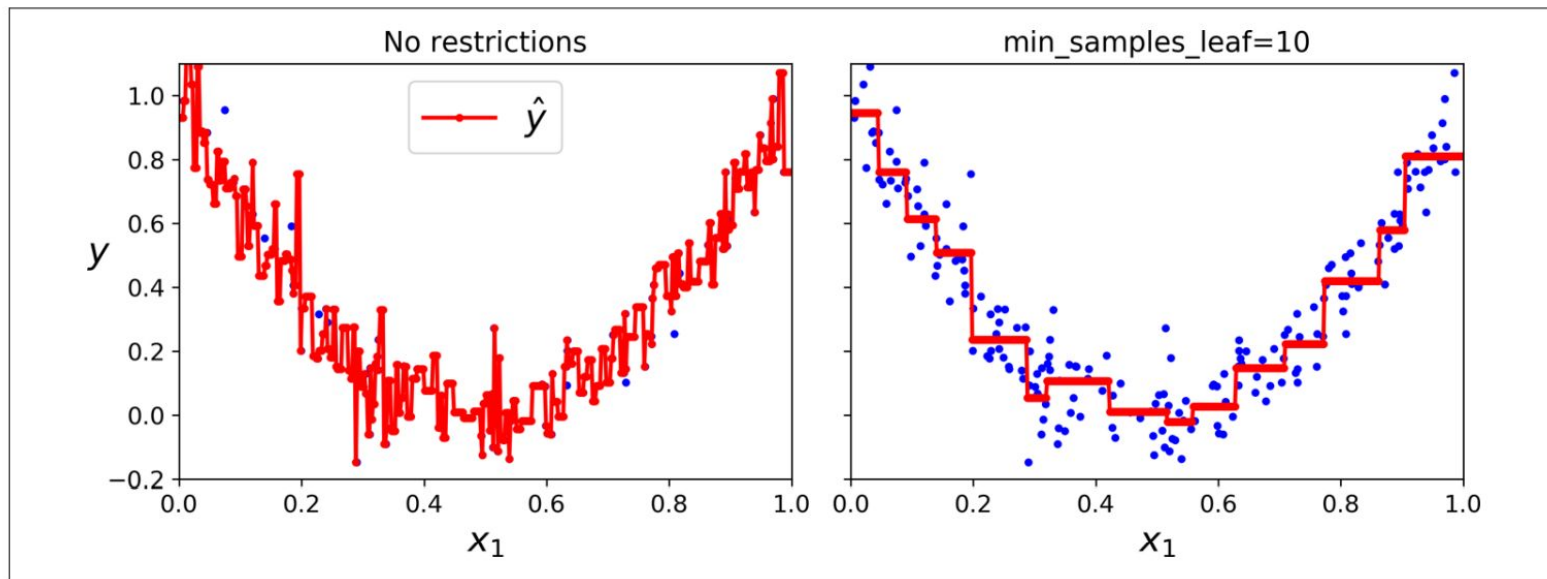


Figure 6-6. Regularizing a Decision Tree regressor

Instabilidade

- Sensíveis a rotações
- Divisões ortogonais
- Utilizar Principal Component Analysis (PCA)
- Instáveis a pequenas variações nos dados treinamento
- Random forests podem auxiliar nessas limitações

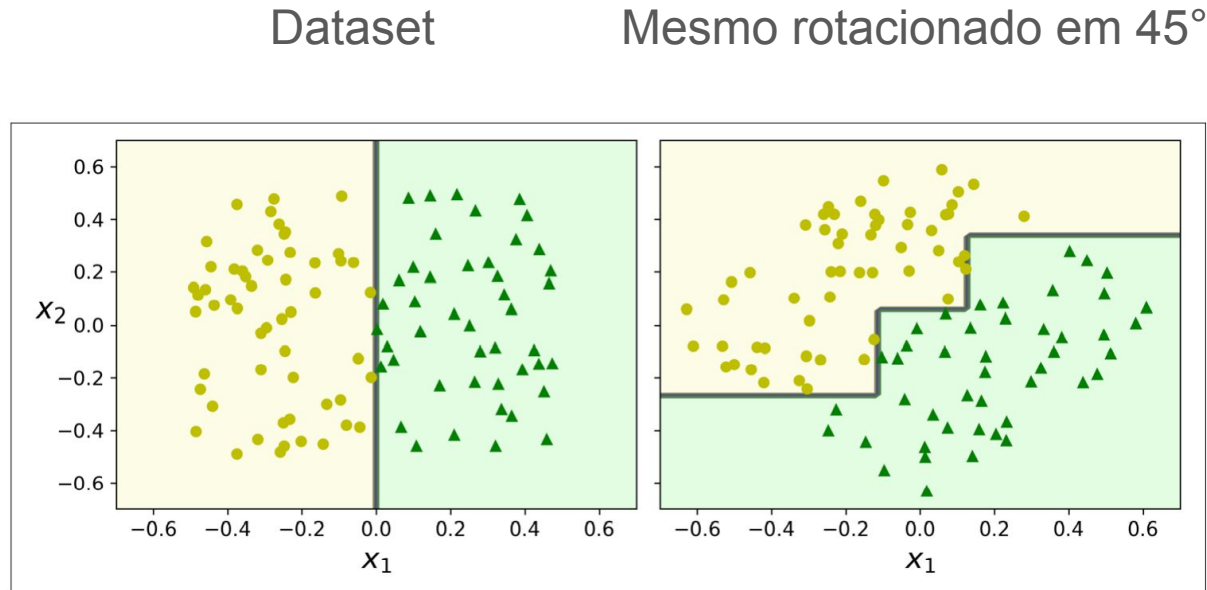
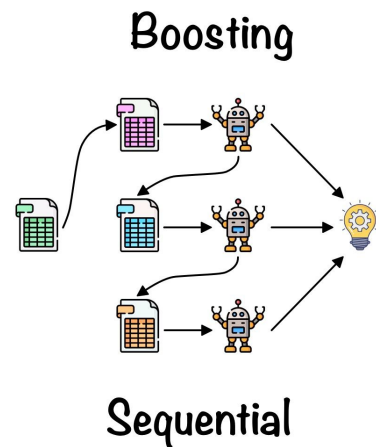
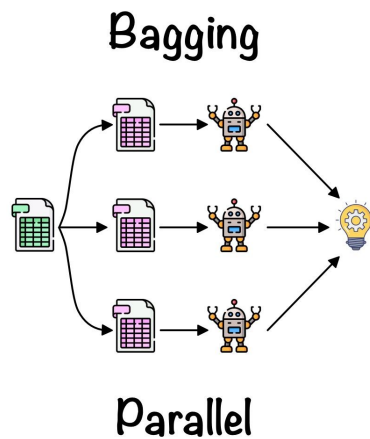


Figure 6-7. Sensitivity to training set rotation

Ensemble learning

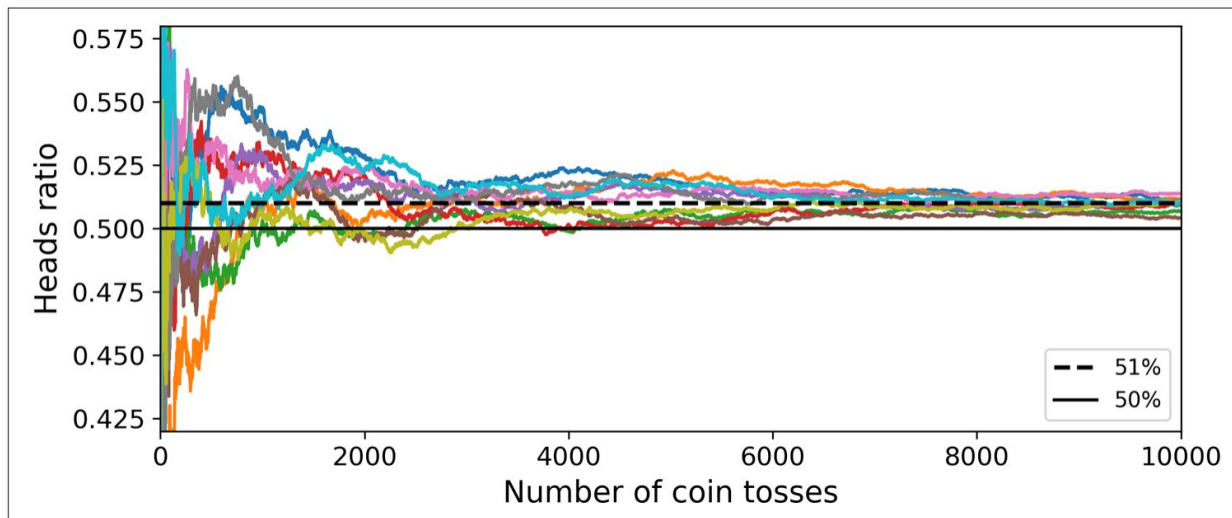
Conceitos gerais

- Agregar respostas
- Inteligência do coletivo
 - Melhor que os métodos individuais
- Tipos de organização
 - Bagging e Pasting
 - Boosting
- Normalmente métodos ensemble obtém os melhores resultados
- Quanto mais independente os modelos melhores os resultados



Poder dos weak learners

- Como modelos simples obtém excelentes resultados?
- Exemplo: jogar uma moeda que cai 51% da vezes em cara
- Com o aumento do número de jogadas, todas as séries convergem (Law of Large Numbers)



Poder dos weak learners

- De maneira similar, com 1000 classificadores independentes que estão corretos 51% do tempo (pouco melhor que acaso) é possível obter até 75% de acurácia
- 97% de acurácia para 10,000 classificadores

OBS: em condições reais os preditores/dados não são totalmente independentes, então não é possível chegar nesse valor, mas sim se aproximar dele.

Votação de classificadores

- Tipos unificação de dados
 - Hard classifier:** maior quantidade de votos
 - Soft classifier:** Leva em consideração a probabilidade/certeza sobre as previsões
 - Stacking:** Treina um modelo para agregar os dados dos diversos modelos da melhor forma

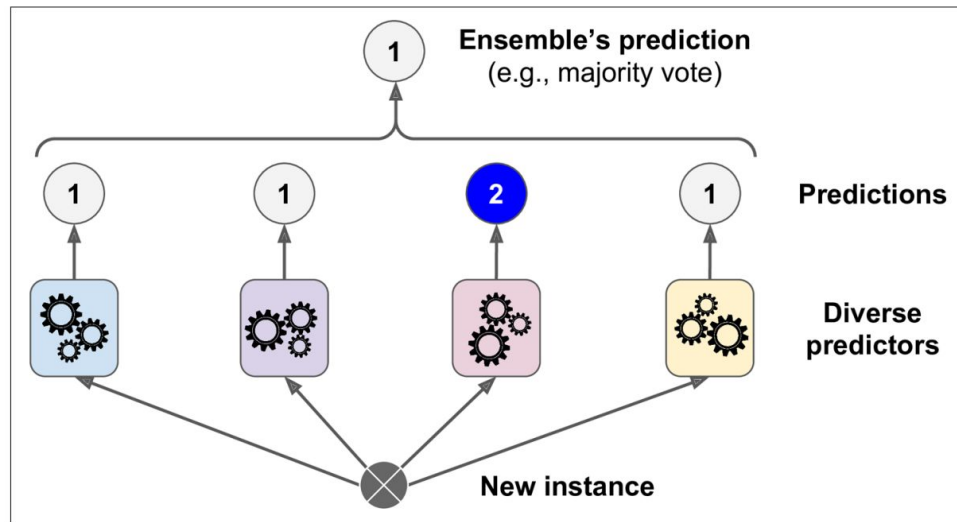


Figure 7-2. Hard voting classifier predictions

Bagging and Pasting

Treinar com o mesmo dataset, mas em subconjuntos randômicos do mesmo.

- **Bagging** - pode utilizar uma instância mais de uma vez
- **Pasting** - somente utiliza uma instância uma vez

OBS:

- Esses modelos podem ser treinados em paralelo, escalam facilmente.
- Geralmente bagging tem melhores resultados, mas não em todos os casos.

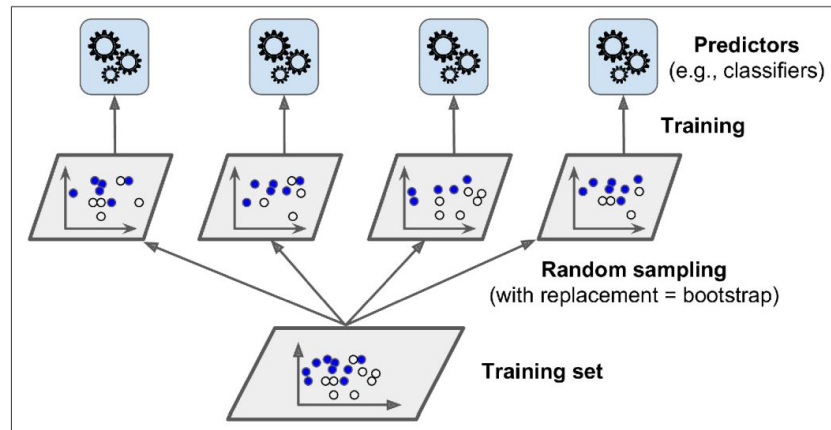
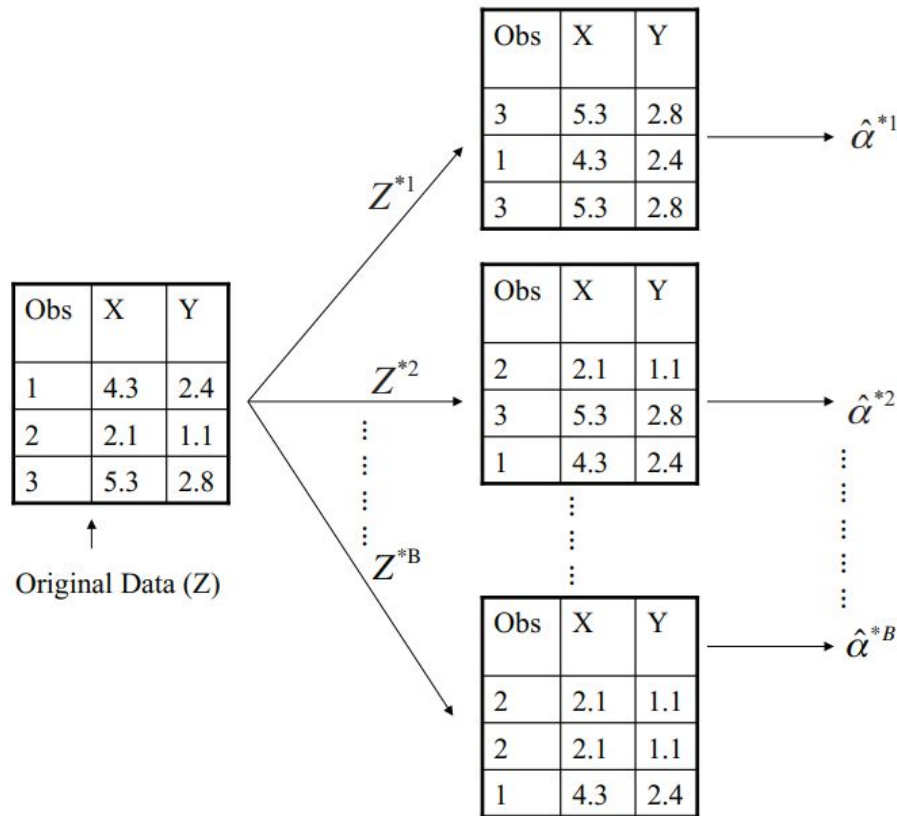


Figure 7-4. Bagging and pasting involves training several predictors on different random samples of the training set

Exemplo de bagging

- Percebe-se a possível repetição de instâncias em alguns subconjuntos



Bagging and Pasting

Cada predictor têm viés e variância maior do que com treinamento no conjunto todo, porém a agregação reduz o viés para valores similares ao do conjunto completo, mas a variância tende a ser menor.

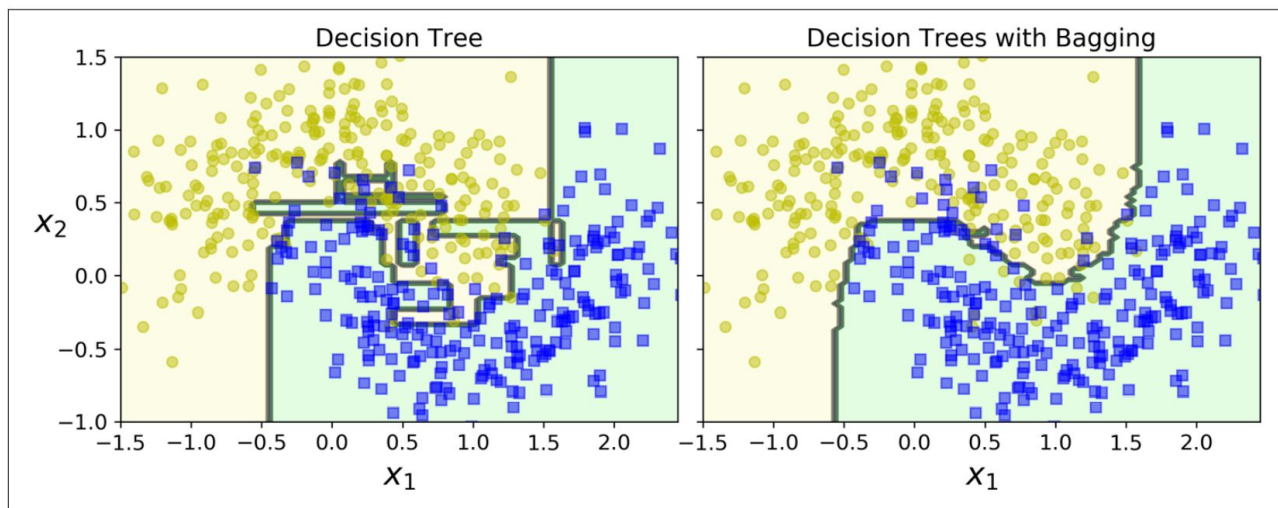


Figure 7-5. A single Decision Tree (left) versus a bagging ensemble of 500 trees (right)

Bagging and Pasting

Out-of-bag evaluation

- Com bagging algumas instâncias podem ser utilizadas diversas vezes em um mesmo preditor e outras não são utilizadas
- O grupo de instâncias separadas tende a ser diferente para cada preditor
- Essas instâncias (OoB) podem ser utilizadas para estimar o resultado do modelo para cada preditor como conjunto de validação

Amostragem de atributos

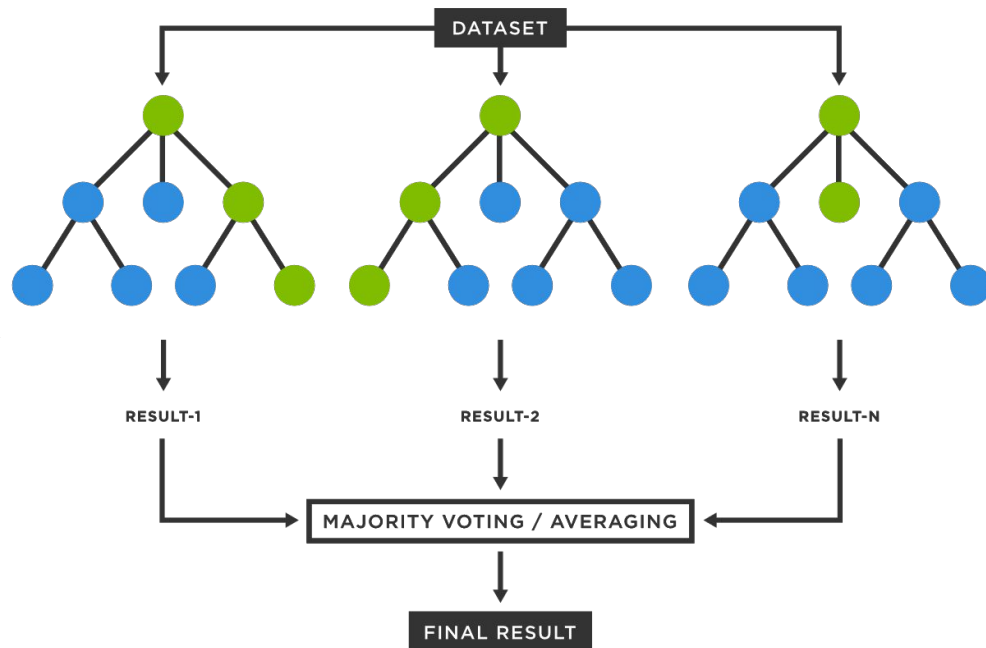
- Tem o conceito similar a escolha aleatória, mas é aplicada aos atributos ao invés das amostras
 - Aplicado aos atributos e amostras: random patches
 - Aplicado somente aos atributos: random subspaces
- Muito útil quando trabalhando com datasets com muitas dimensões, como imagens por exemplo.
- Busca reduzir o overfitting

Random Forests

Aplica os conceitos discutidos até aqui

- Usualmente utiliza bagging
- Agregado de árvores de decisão
- Ao invés de procurar o melhor atributo para dividir um nó, utiliza a divisão em um conjunto aleatório de atributos

OBS: reduz o overfitting em relação às árvores de decisão



Extremely Randomized Trees (Extra Trees)

- É possível aumentar ainda mais a aleatoriedade escolhendo thresholds aleatórios para cada atributo, ao invés de buscar o melhor como é feito nas árvores de decisão

OBS: Além disso, o treinamento é bem mais rápido, pois escolher o melhor threshold é uma das atividades mais complexas do treinamento das árvores de decisão.

Importância de cada atributo

A floresta aleatória é um bom modelo para demonstrar quais são os atributos mais importantes para as previsões, interessante para realizar a escolha de atributos.

Very important

Not important

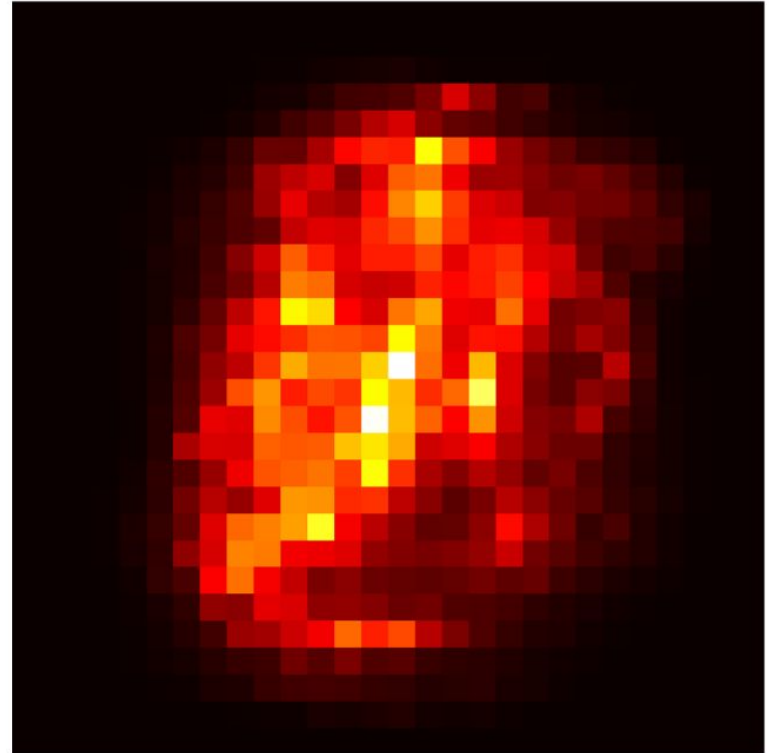
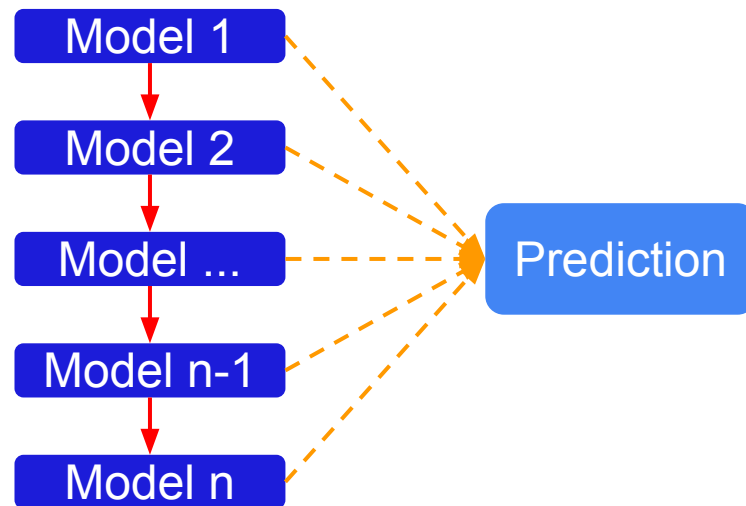


Figure 7-6. MNIST pixel importance (according to a Random Forest classifier)³⁵

Boosting

- Similar aos anteriores, mas ao invés de paralelo trabalha os dados sequencialmente
- Não podem ser treinados paralelamente
- Em alguns modelos altera a importância de parte dos dados para cada uma das instâncias



AdaBoost

- Realiza predição com pesos iguais para as amostras
- Aquelas que foram classificadas incorretamente recebem maior peso no próximo classificador
- Repete-se o processo
- Agrega-se o resultado final

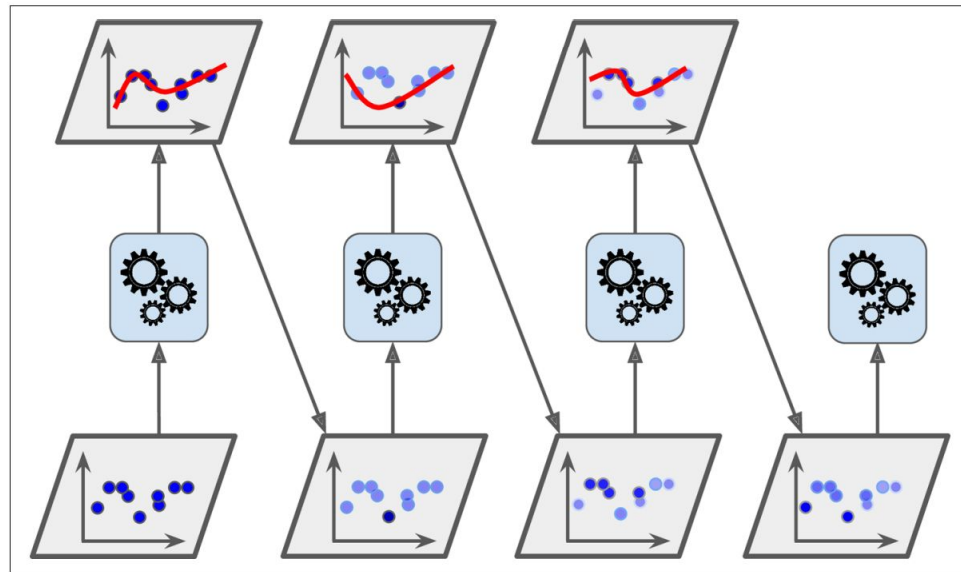
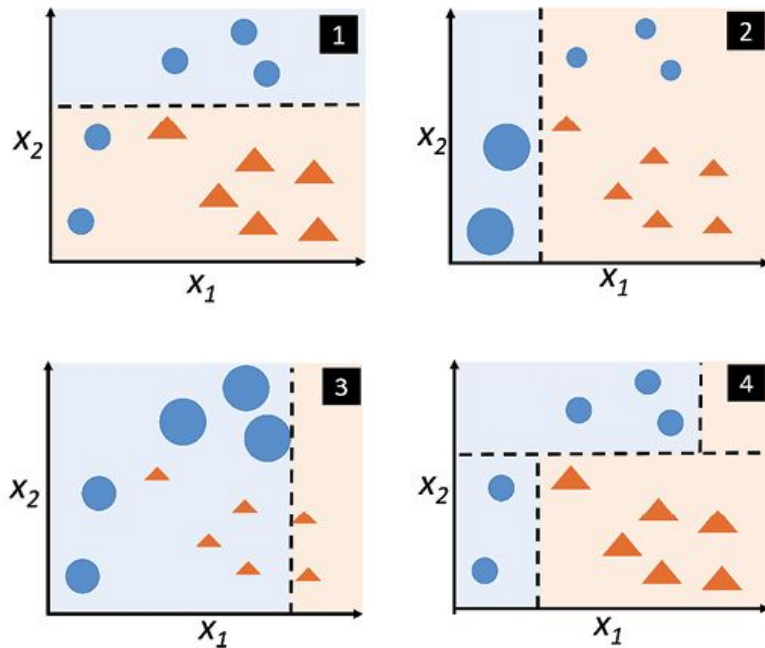


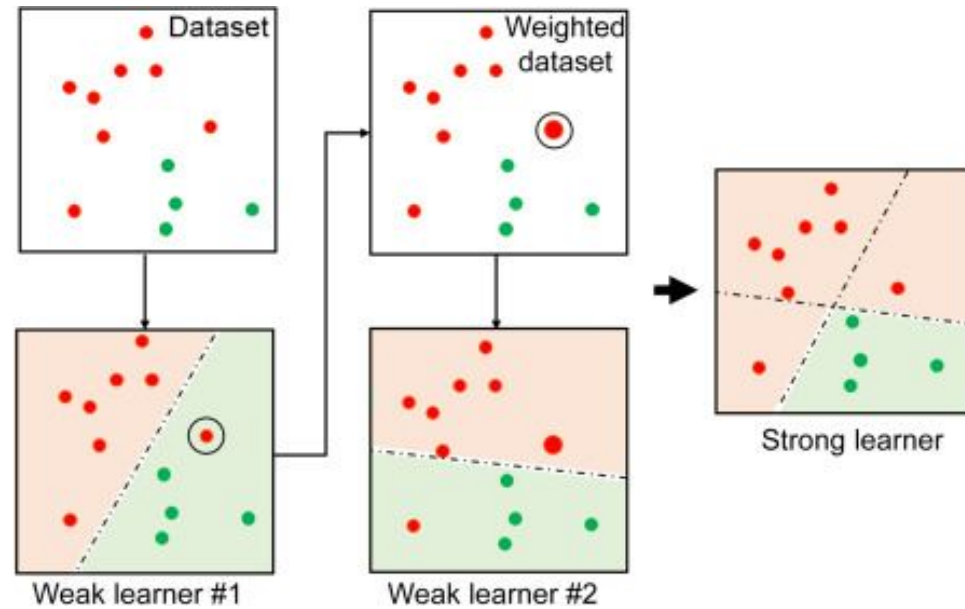
Figure 7-7. AdaBoost sequential training with instance weight updates

AdaBoost

Exemplo 1

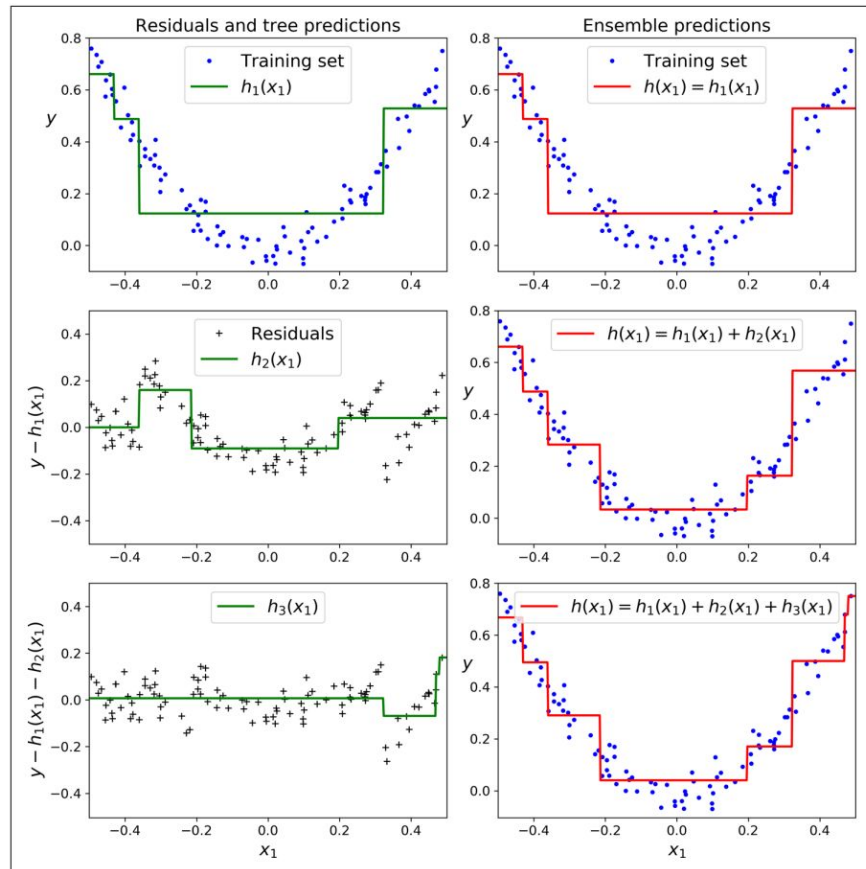


Exemplo 2



Gradient Boosting

Similar ao AdaBoost, mas ao invés de alterar os pesos das instâncias, busca-se treinar o modelo com base no erro residual da predição anterior.



Gradient Boosting

- Comparação entre muitos e poucos preditores
- Parada antecipada

OBS: é possível treinar aleatoriamente em uma subparcela do grupo de treinamento, reduzindo overfitting. Isso pode reduzir consideravelmente o tempo de treinamento e é conhecido como Stochastic Gradient Boosting.

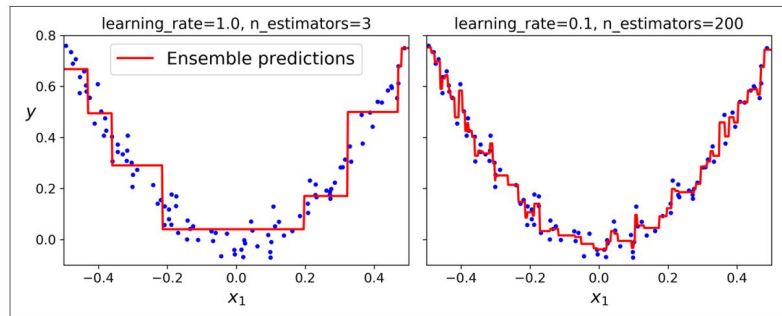


Figure 7-10. GBRT ensembles with not enough predictors (left) and too many (right)

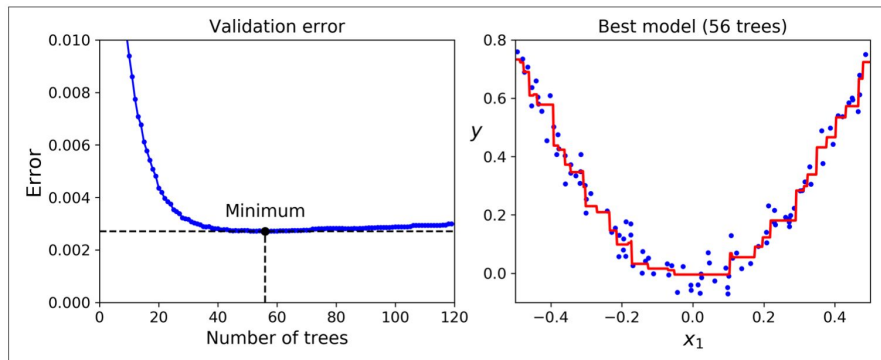


Figure 7-11. Tuning the number of trees using early stopping

Stacking

Ao invés de usar hard ou soft voting, treina-se um modelo para agregar as previsões do grupo

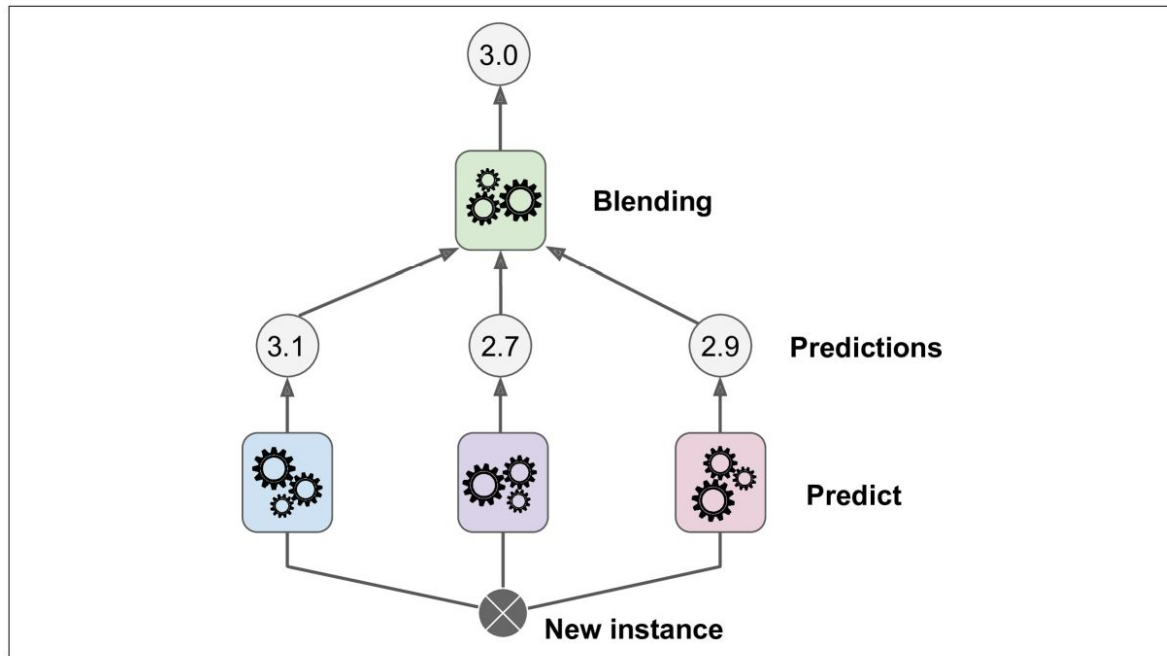


Figure 7-12. Aggregating predictions using a blending predictor

Stacking

Primeiro treina-se os modelos individualmente em um subset do dataset inicial

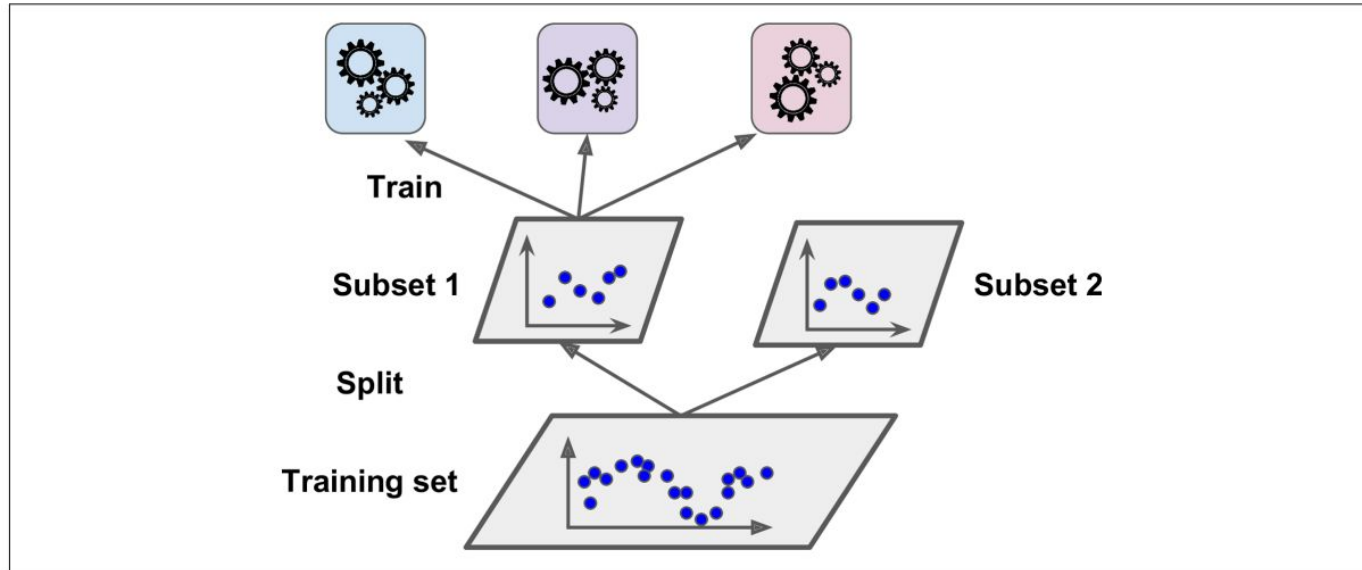


Figure 7-13. Training the first layer

Stacking

Na sequência treina-se no outro subset o blender ou agregador.

- Tende-se a ter resultados melhores utilizando diversos tipos de modelos nas previsões

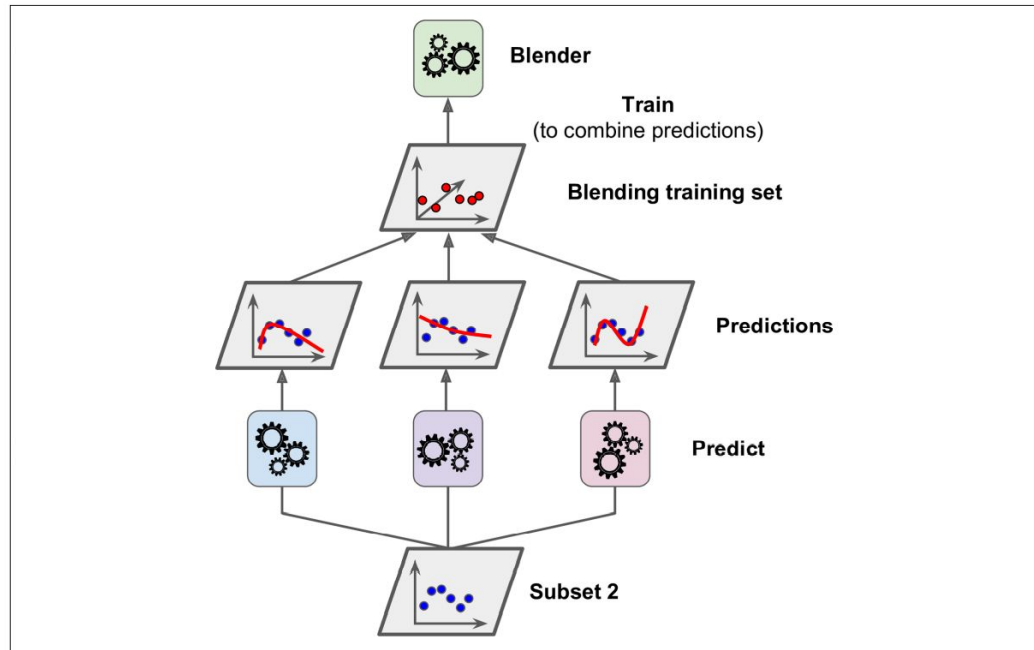


Figure 7-14. Training the blender