

Métodos de Conjunto

November 11, 2025

0.1 Métodos de Conjunto em Aprendizado de Máquina

Modelos de Conjunto em Aprendizado de Máquina

Vamos imaginar um gerente musical participando de uma competição internacional. Ele tem acesso a uma grande variedade de músicos com diferentes especialidades:

- Músicos clássicos com a habilidade de compor peças tradicionais.
- Músicos eletrônicos que são especialistas em usar instrumentos eletrônicos.
- Músicos de jazz com um grande senso de improvisação.
- Músicos solistas que podem executar solos complexos e destacar suas habilidades técnicas.

Dada a ampla gama de especialidades e experiências musicais, o gerente pode combinar todos eles para criar uma performance única e memorável.

Pense nos modelos de conjunto como uma orquestra de músicos, onde cada pessoa se especializa em um instrumento específico, como piano, trompete, bateria e mais. A combinação dessas habilidades cria uma melodia harmoniosa.

O aprendizado em conjunto usa a mesma lógica:

Ele combina múltiplos algoritmos para obter um desempenho preditivo melhor do que o de um único modelo. Não há um número predefinido de modelos a considerar, e alguns objetivos de negócios podem exigir mais modelos do que outros.

0.2 Erro do Modelo e Redução desse Erro com Conjuntos

O erro que surge de qualquer modelo de máquina pode ser dividido em três componentes matematicamente:

$$\text{Viés} + \text{Variância} + \text{Erro Irreduzível}$$

Por que isso é importante no contexto atual? Para entender o que acontece por trás de um modelo de conjunto, você precisa primeiro saber o que causa um erro no modelo.

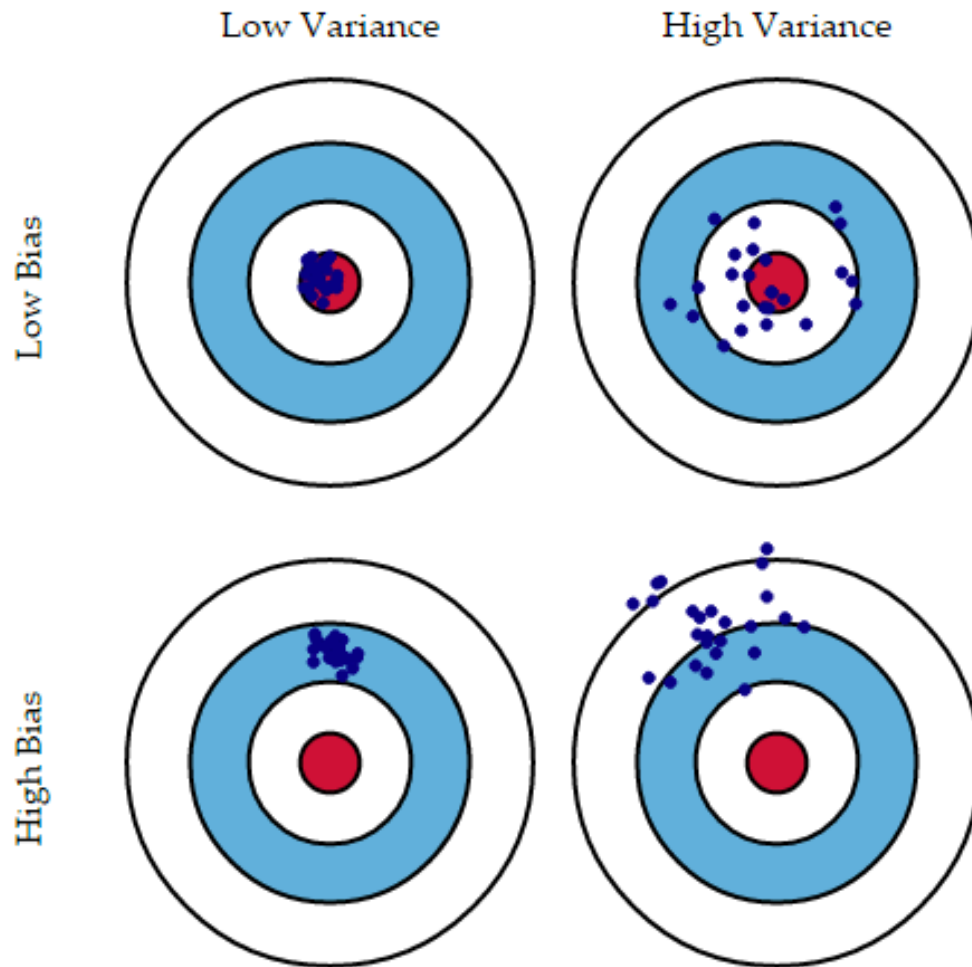
Vamos analisar esses erros:

Erro de Viés: Isso é útil para quantificar o quanto, em média, os valores previstos são diferentes do valor real. Um alto erro de viés significa que temos um modelo com baixo desempenho que continua perdendo tendências essenciais.

Variância: Por outro lado, a variância quantifica como as previsões feitas na mesma observação diferem. Um modelo com alta variância se ajustará excessivamente à sua população de treinamento e terá um desempenho ruim em qualquer observação além do treinamento. O diagrama a seguir dará mais clareza (assuma que o ponto vermelho é o valor real e os pontos azuis são previsões):

```
[1]: from IPython.display import Image  
Image(filename='Variancia_Vies.png')
```

[1]:



Tipicamente, à medida que você aumenta a complexidade do seu modelo, verá uma redução no erro devido ao menor viés no modelo. No entanto, isso só acontece até um certo ponto. À medida que você continua a tornar seu modelo mais complexo, acaba ajustando-o excessivamente, e assim seu modelo começará a sofrer de alta variância.

0.3 Tipos de Métodos de Conjunto

Existem diferentes tipos de métodos de conjunto, e cada um traz um conjunto de vantagens e desvantagens. Esta seção cobre esses aspectos para ajudá-lo a fazer a escolha certa para seus casos de uso.

Antes de mergulhar em cada método, vamos entender o que são meta e base learners para uma melhor compreensão dos próximos conceitos.

1. **Base learners:** São o primeiro nível de uma arquitetura de aprendizado em conjunto, e cada um deles é treinado para fazer previsões individuais.
2. **Meta learners:** Estão no segundo nível e são treinados com base na saída dos base learners.

0.4 Bagging

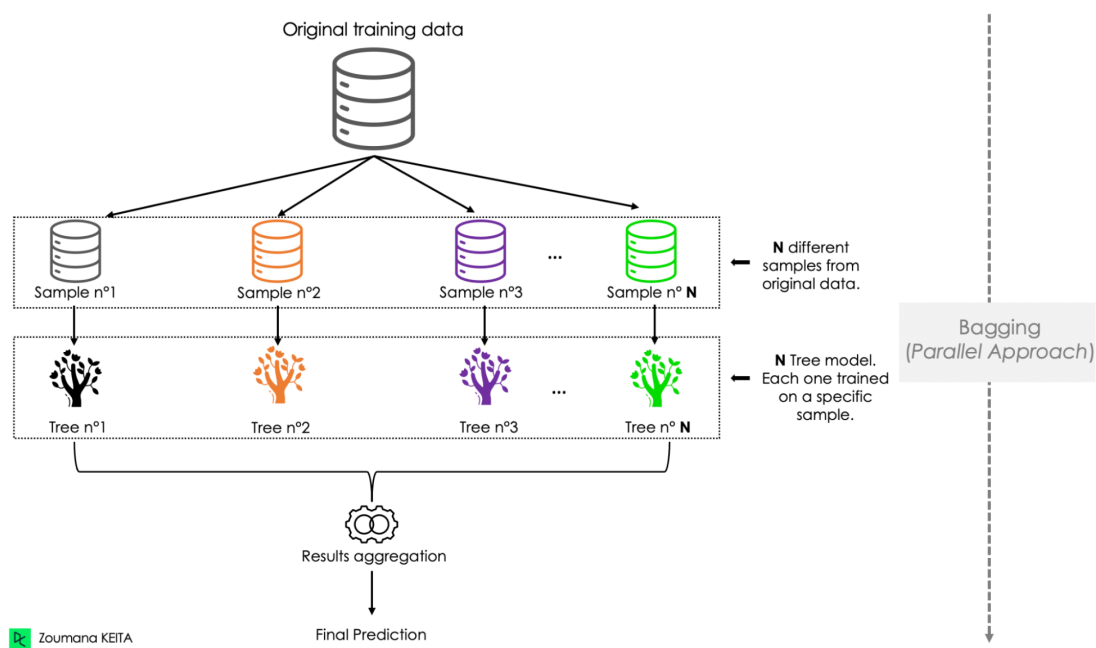
Bagging também é conhecido como agregação bootstrap. Esta técnica é semelhante à floresta aleatória, mas usa todos os preditores, enquanto a floresta aleatória usa apenas um subconjunto de preditores em cada árvore.

No bagging, uma amostra aleatória de dados do conjunto de treinamento é selecionada com reposição, o que permite a duplicação de instâncias de amostra em um conjunto. Abaixo estão as principais etapas envolvidas no bagging:

- Geração de múltiplas reamostragens bootstrap.
- Execução de um algoritmo em cada reamostragem para fazer previsões.
- Combinação das previsões, tomando a média das previsões ou a maioria dos votos (para classificação).

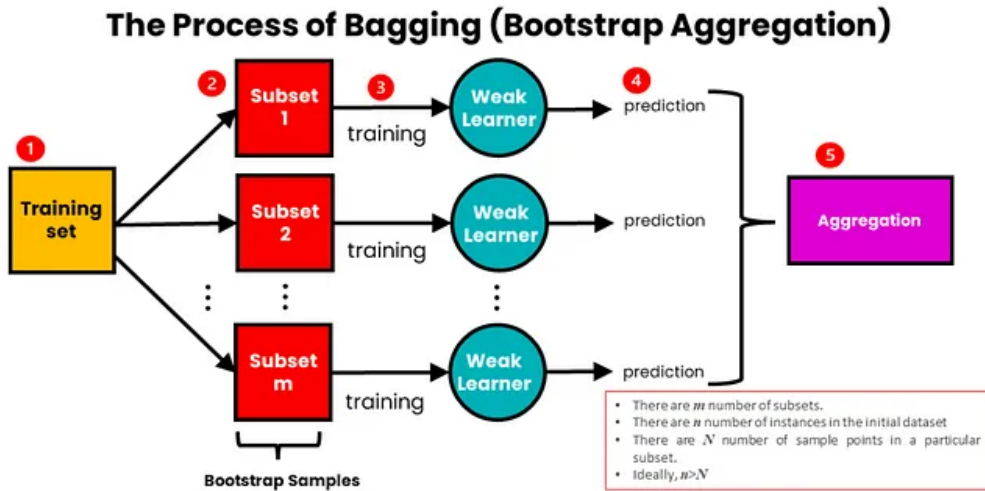
```
[2]: Image(filename='Bagging.png')
```

[2]:



```
[3]: Image(filename='Bagging2.png')
```

[3]:



0.4.1 Vantagens

- O processo de modelagem é direto e não requer conceitos matemáticos profundos, além de poder lidar com valores ausentes.
- O pacote scikit-learn facilita a implementação da lógica subjacente. Ele contém todos os módulos para combinar as previsões de cada modelo, também conhecidos como base learners.
- Tem um efeito significativo na redução da variância em classificadores de alta variância. Isso é útil ao lidar com dados de alta dimensão, evitando que o modelo generalize com precisão em novos dados.
- O bagging fornece uma estimativa imparcial do erro out-of-bag, que corresponde ao erro/perda média que todos esses classificadores produzem.

0.4.2 Desvantagens

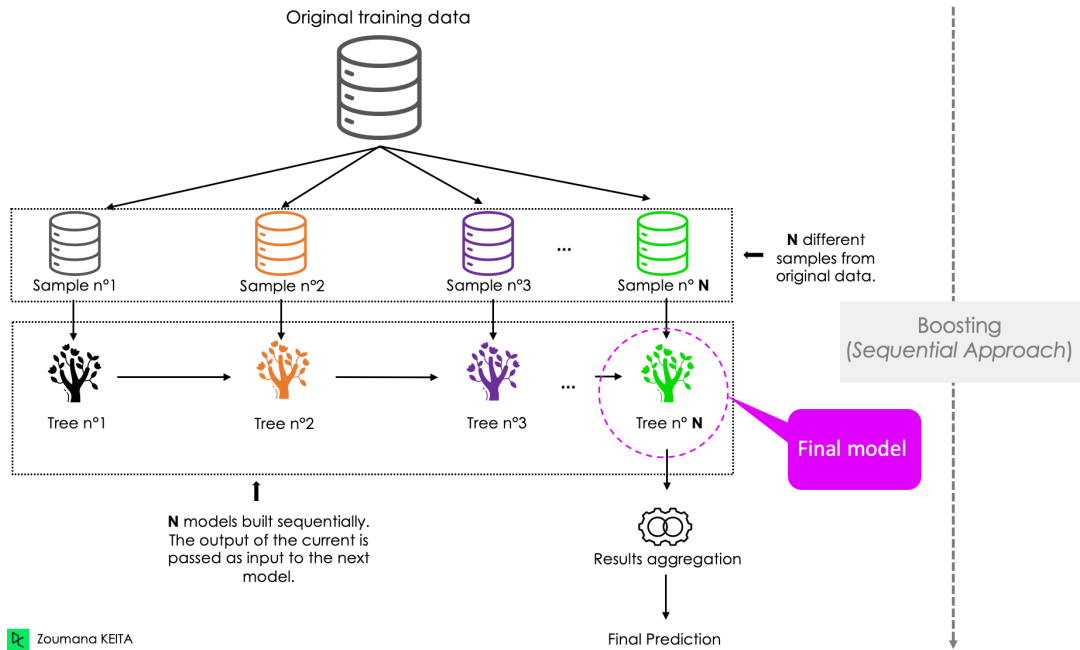
- O bagging é computacionalmente caro devido ao uso de vários modelos.
- A média envolvida nas previsões torna difícil interpretar o resultado final.

0.5 Boosting

O boosting adota uma abordagem sequencial, onde a previsão do modelo atual é transferida para o próximo. Cada modelo foca iterativamente nas observações que foram classificadas incorretamente por seus predecessores. Abaixo está o processo geral:

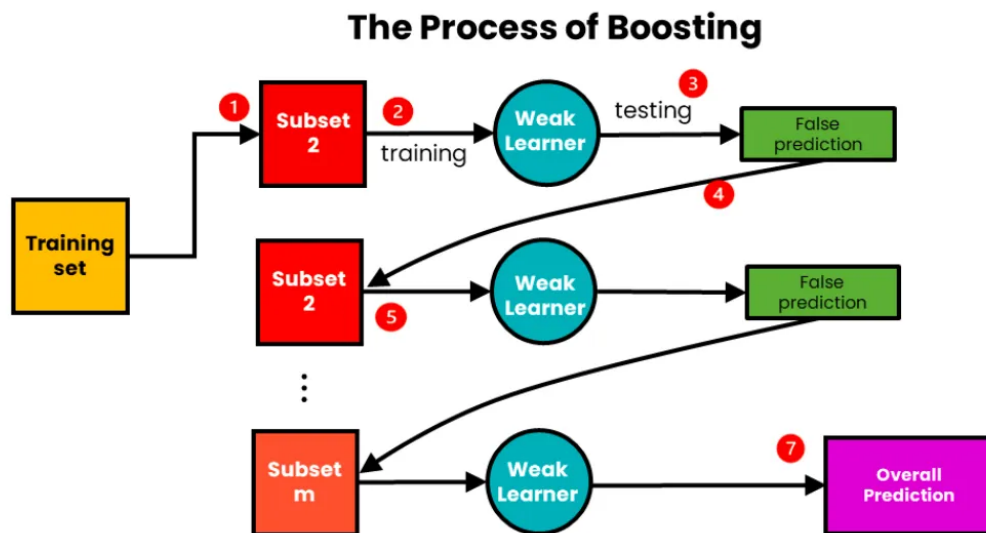
```
[4]: Image(filename='Boosting.png')
```

```
[4]:
```



```
[5]: Image(filename='Boosting2.png')
```

```
[5]:
```



0.5.1 Vantagens

- Assim como o bagging, o boosting é fácil de entender e implementar. Além disso, não requer pré-processamento e pode lidar com valores ausentes nos dados.
- Reduz eficientemente o viés.
- Os algoritmos de boosting priorizam características que aumentam a precisão geral durante o treinamento. Esse processo reduz a dimensionalidade dos dados, reduzindo assim o tempo de computação.

0.5.2 Desvantagens

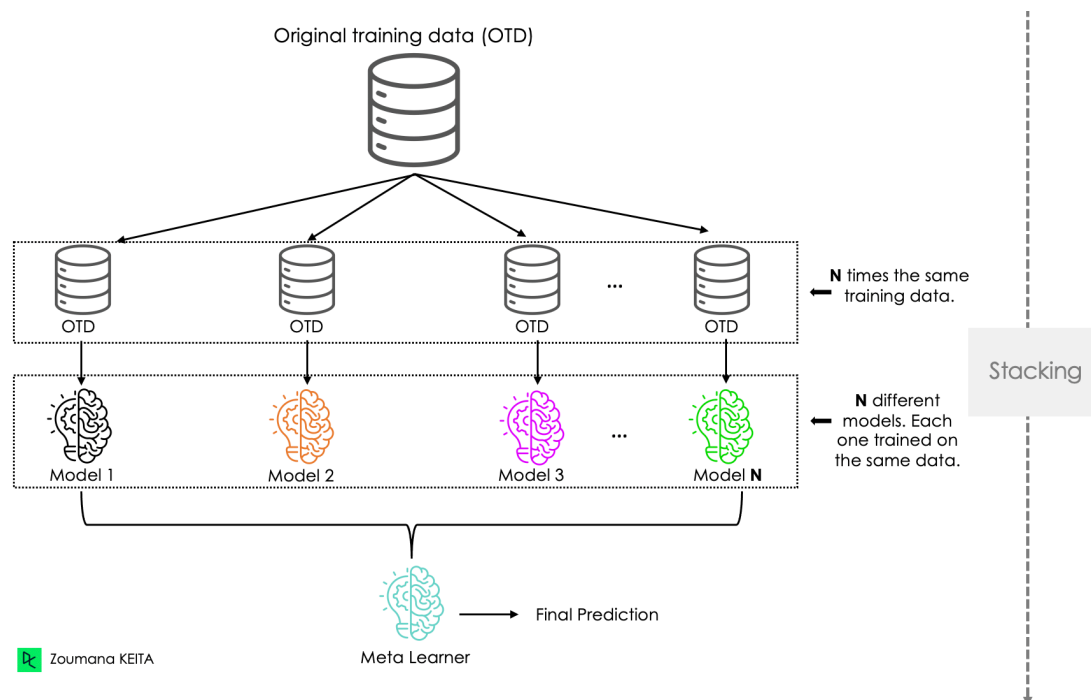
- A abordagem sequencial do boosting faz com que os próximos modelos corrijam os erros de seu predecessor. Isso torna o modelo geral vulnerável a outliers.
- O boosting não é escalável pelo mesmo motivo do aspecto sequencial.

0.6 Stacking

O stacking é bastante semelhante ao boosting. As previsões dos base learners são empilhadas e usadas como entrada para treinar o meta learner para produzir previsões mais robustas. O meta learner é então usado para fazer previsões finais. Bagging e boosting tipicamente usam base learners homogêneos, enquanto o stacking tende a incluir base learners heterogêneos.

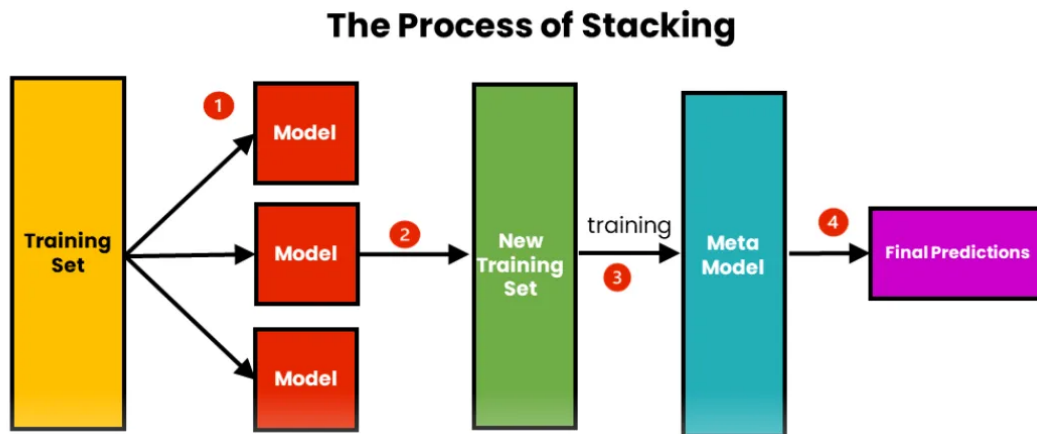
```
[6]: Image(filename='Stacking.png')
```

[6]:



```
[7]: Image(filename='Stacking2.png')
```

[7]:



0.6.1 Vantagens

- Aproveita as forças de múltiplos modelos de alto desempenho para tarefas de classificação e regressão.
- Assim como outros modelos de conjunto, o stacking ajuda a construir um modelo mais preciso do que os modelos individuais usados isoladamente.

0.6.2 Desvantagens

- Usar modelos básicos complexos pode aumentar o risco de overfitting.
- Diferentes níveis de treinamento de um modelo podem tornar a arquitetura de stacking complexa de implementar.

0.7 Blending

O blending é semelhante ao stacking. No blending, a estrutura dos dados é composta por dados de treinamento, hold-out e teste. Os meta learners são treinados nos dados de treinamento. Em seguida, suas previsões são combinadas com os dados hold-out para construir o modelo meta final, que usa os dados de teste para fazer as previsões finais.

0.7.1 Vantagens

- Assim como o stacking e muitos outros métodos de conjunto, o blending pode aumentar o desempenho do modelo final em muitos casos.
- O uso dessa técnica tem sido bem-sucedido em vencer muitas competições.

0.7.2 Desvantagens

- Dividir os dados originais de acordo com a arquitetura de blending pode limitar o uso de dados para treinar os modelos base, resultando em desempenho ruim.

- Assim como outros modelos de conjunto, a interpretabilidade do modelo final é reduzida devido ao aumento da complexidade, dificultando a obtenção de insights cruciais para os negócios.

0.8 Voting

No voting, múltiplos modelos são treinados independentemente, e suas previsões são combinadas para fazer uma previsão final usando votação rígida, votação suave ou votação ponderada:

- Na votação rígida, a previsão final é a previsão mais comum de todos os modelos.
- Na votação suave, cada modelo gera uma distribuição de probabilidade em vez de uma previsão binária. Então, a classe com a maior probabilidade é a prevista.
- Finalmente, na votação ponderada, assume-se que alguns modelos têm mais habilidade do que outros, e esses modelos são atribuídos com mais contribuição ao fazer previsões.

Vantagens - A arquitetura de voting é simples de implementar em comparação com o stacking e o blending. Além disso, não requer ajustes complexos.

- Usar múltiplos base learners no voting torna-o menos suscetível à influência de modelos individuais, o que contribui para previsões mais estáveis e confiáveis.

Desvantagens - Pode ser difícil lidar com o conflito de previsões dos modelos, o que torna difícil tomar a decisão final de maneira significativa.

- Adicionar mais modelos ao modelo de voting em conjunto não melhora necessariamente o desempenho final.

0.9 Cascading

O cascading usa uma abordagem de stacking, mas com apenas um modelo em cada camada. O primeiro modelo é treinado em todos os dados de treinamento, e o próximo modelo é treinado na saída do modelo anterior. O objetivo de usar a estratégia é aprender padrões complexos dos dados, permitindo que o modelo faça previsões melhores.

0.9.1 Vantagens

- O próximo modelo é especializado na saída do modelo anterior, o que reduz o ruído nos dados.
- Mais robusto ao overfitting e pode ter um bom desempenho em dados do mundo real.

0.9.2 Desvantagens

- Implementar o cascading pode ser complexo, pois envolve treinar cada um dos modelos na sequência.
- Obter a arquitetura de cascading ideal pode exigir muita experimentação e ajuste fino.

0.10 Visão Geral dos Algoritmos de Conjunto

As seções anteriores cobriram os diferentes tipos de modelos de conjunto. Agora, vamos ter uma visão geral de alguns modelos populares.

0.10.1 Random Forest

A Random Forest é um modelo comumente usado que pode resolver problemas de classificação e regressão. Uma floresta aleatória é composta por muitas árvores de decisão que são treinadas usando bagging. Seu resultado é determinado pela média das previsões das árvores individuais.

O tamanho do nó, o número de árvores e o número de características amostradas são os três principais hiperparâmetros que precisam ser definidos antes de treinar a floresta aleatória.

0.10.2 XGBoost

O Extreme Gradient Boosting, ou XGBoost, é usado tanto para classificação quanto para regressão. O XGBoost é projetado para ser escalável e altamente eficiente, e implementa a estrutura de árvores de decisão de boosting de gradiente.

É adequado para processar conjuntos de dados em grande escala e é compatível com os principais ambientes distribuídos, como Hadoop, MPI (Message Passing Inference) e SGI (Sun Grid Engine).

0.10.3 AdaBoost

O Adaptive Boosting, ou AdaBoost, é um dos primeiros classificadores de boosting em conjunto para algoritmos de boosting bem-sucedidos para classificação binária. É adaptativo, pois os pesos são reatribuídos a cada instância, e as instâncias classificadas incorretamente são atribuídas com pesos mais altos.

[]: