

Gradient Boosting

Adailton, Rennan

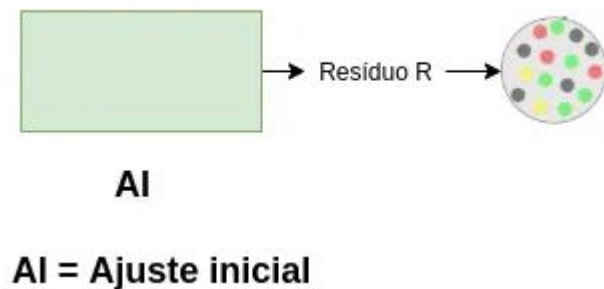
A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

O que é?

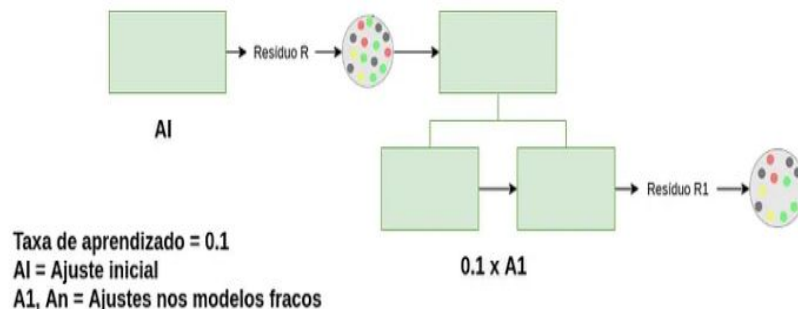
- É uma técnica de aprendizado de máquina para problemas de regressão e classificação,
- que produz um modelo de previsão na forma de um conjunto de modelos de previsão fracos.
- É montado o modelo em etapa, focando nos erros e permitindo a otimização de uma função.
- O objetivo do algoritmo é criar uma corrente de modelos fracos, onde cada um tem como objetivo minimizar o erro do modelo anterior, por meio de uma função de perda.

Como funciona o algoritmo?

Um primeiro modelo é criado com uma aproximação simples. Dessa aproximação, temos o resíduo, que é a distância entre o que foi previsto e o valor real:

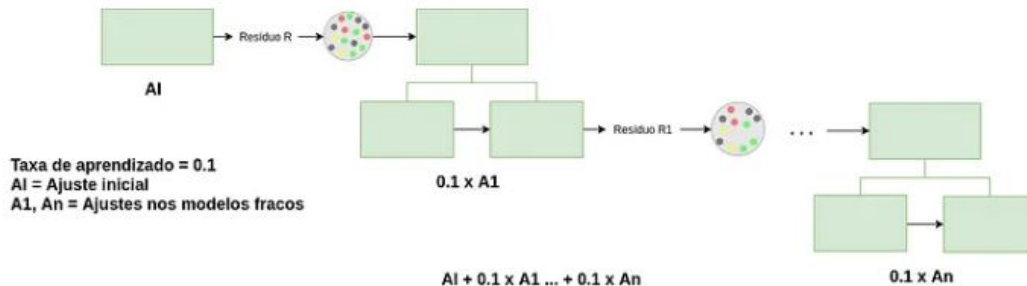


Um próximo modelo é criado e ajustado em cima do resíduo gerado pelo modelo anterior. Em seguida, um novo resíduo é calculado com esse segundo modelo:



Como funciona o algoritmo?

Essas interações são repetidas por um determinado número de vezes, buscando minimizar o resíduo gerado pelos modelos fracos, ou seja, até que a distância entre o previsto e o valor real seja o menor possível. O modelo final é a soma dos ajustes de todos os modelos fracos:



Exemplo do algoritmo utilizando o Scikit-Learn

O Scikit-Learn dispõe de uma implementação do algoritmo Gradient Boosting preciso e eficaz para a resolução de problemas de classificação e regressão.

De acordo com a documentação do Scikit-Learn, os parâmetros mais importantes na concepção do modelo são:

- **loss:** A função de perda a ser otimizada durante o treinamento.
 - Para problemas de **regressão**, as opções comuns são '**squared_error**' '**absolute_error**'. Para problemas de **classificação**, as opções comuns incluem '**deviance**' e '**exponential**'.
- **learning_rate:** Taxa de aprendizado que determina a importância de cada árvore na concepção do modelo final e na minimização do resíduo gerado. Deve ser um valor no intervalo de 0.0 a 1.
- **n_estimators:** Número de árvores ou estágios utilizados na construção e treinamento do modelo.
- Também é possível controlar o tamanho de cada árvore através dos parâmetros:
- **max_depth:** A profundidade máxima da árvore.

Existem outros parâmetros que podem ser observados na [documentação](#) oficial.

Bora codar!