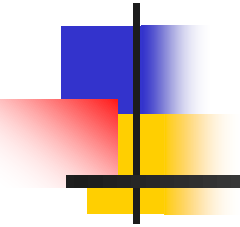


GRADIENT BOOSTING MACHINES (GBM)



GRADIENT BOOSTING MACHINES - GBM



O algoritmo Gradient Boosting é um dentro do grupo de classificadores *Ensemble*. Esses classificadores são métodos que utilizam uma combinação de resultados de preditores fracos, com o objetivo de produzir um melhor modelo preditivo. Os preditores fracos são modelos que, quando utilizados individualmente possuem uma acurácia abaixo do esperado.

A abordagem envolve a adição sequencial de árvores de decisão, onde cada árvore corrige os erros das anteriores.

GRADIENT BOOSTING MACHINES - GBM



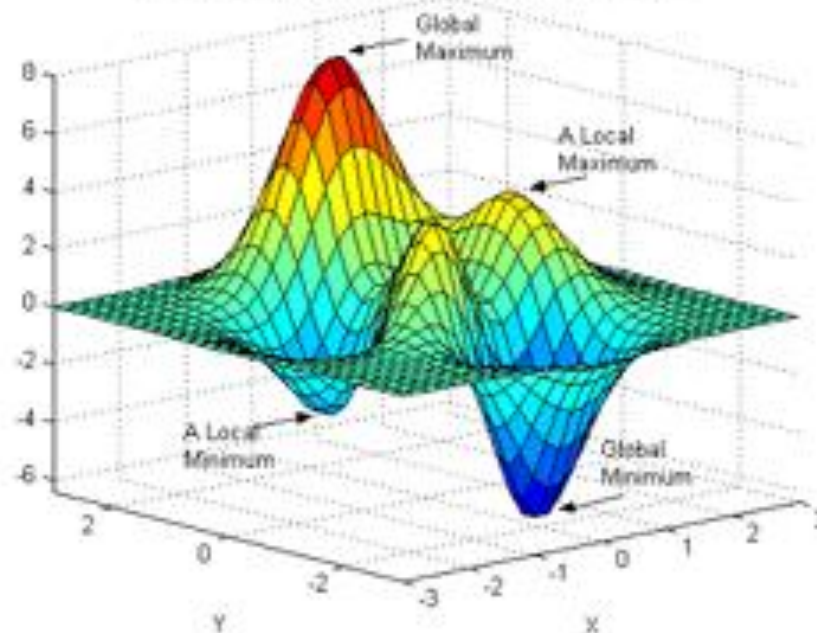
Na técnica de Boosting, cada classificador fraco é treinado com um conjunto de dados, de forma sequencial e de uma maneira adaptativa, onde um modelo base depende dos anteriores, e no final são combinados de uma maneira determinística.

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.

Aos ajustes de cada modelo fraco é multiplicado um valor chamado de taxa de aprendizagem. Esse valor, tem como objetivo determinar o impacto de cada árvore no modelo final. Quanto menor o valor, menor a contribuição de cada árvore.

Gradiente

- Gradiente Descendente é um algoritmo de otimização usado para minimizar algumas funções movendo-se iterativamente na direção da descida mais íngreme, conforme definido pelo negativo do gradiente. Nos modelos de machine learning, usamos gradiente descendente para atualizar os parâmetros do nosso modelo.



Gradiente

- O método do Gradiente Descendente se baseia na derivada parcial da função $f(\mathbf{x})$ para cada valor X , ou seja:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_d} \right]$$

- O Gradiente descendente (GD) é um método para encontrar o mínimo de uma função de forma iterativa:

Algoritmo: Escolha um chute inicial, $\beta^{(0)} \in \mathbb{R}^p$, repita:

$$\beta^{(k+1)} = \beta^{(k)} - \alpha_k \nabla J(\beta^{(k)}), \quad k = 0, 1, \dots$$

pare quando atingir convergência.

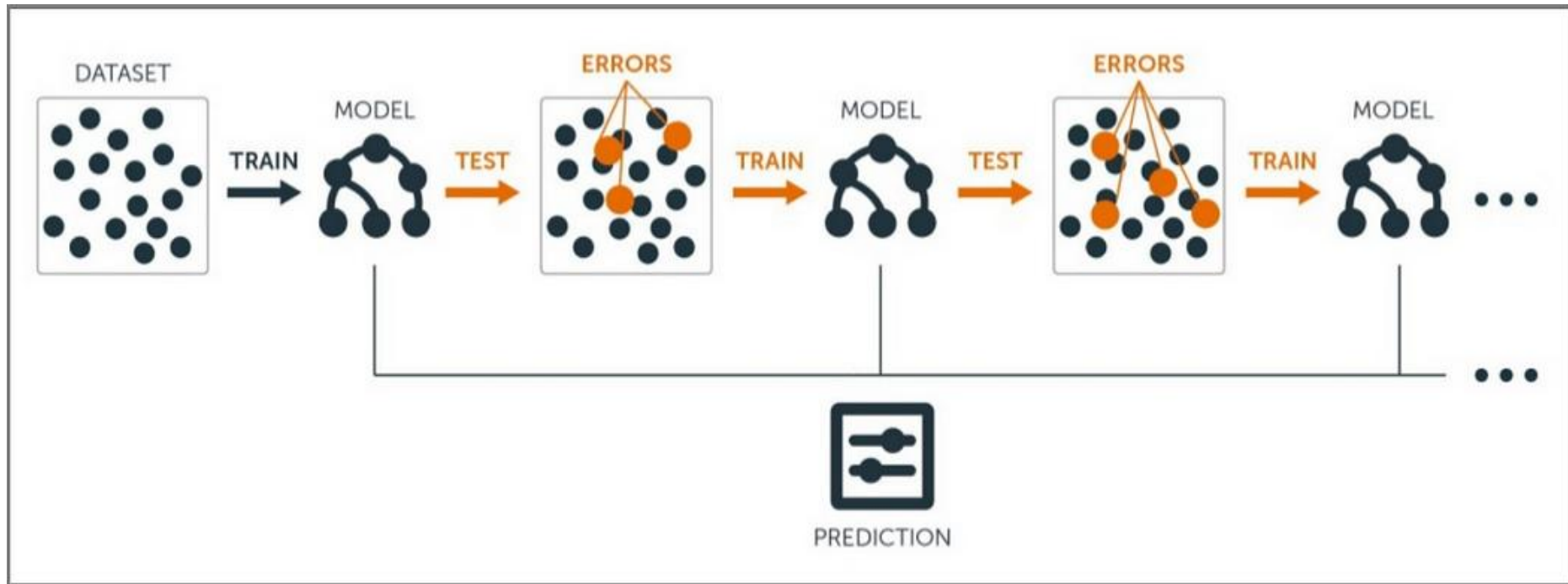
Onde J é a função custo e β são os pesos da minha função

Passo a Passo



1. Começa com uma predição inicial simples (média)
2. Calcula erros (resíduos)
3. Treina nova árvore para prever os resíduos
4. Ajusta predições com learning rate
5. Repete o processo várias vezes

GRADIENT BOOSTING MACHINES - GBM



Funcionamento:

- Inicia-se com uma árvore de decisão simples (modelo fraco).
- As previsões do modelo são calculadas e os erros (resíduos) são identificados.
- Uma nova árvore é então treinada para prever esses resíduos.
- Este processo se repete, somando os modelos, até que a precisão desejada seja atingida.

GRADIENT BOOSTING MACHINES - GBM



Gradient Boosted Decision Trees

$$\hat{y}_i^1 = f_1(x_i)$$



$$f_1(x_i) \rightarrow y_i$$

$$\hat{y}_i^2 = \hat{y}_i^1 + f_2(x_i)$$



$$f_2(x_i) \rightarrow y_i - \hat{y}_i^1$$

$$\hat{y}_i^M = \hat{y}_i^{M-1} + f_M(x_i)$$



$$f_M(x_i) \rightarrow y_i - \hat{y}_i^{M-1}$$

Pontos Fortes e Fracos GBM



PONTOS FORTES

- Precisão: Produz modelos altamente precisos devido à combinação de várias árvores.
- Flexibilidade: Capaz de lidar com diferentes tipos de variáveis e formatos de dados, como dados numéricos e categóricos.
- Regularização: Técnicas como taxa de aprendizado e *subsampling* ajudam a reduzir o overfitting, melhorando a generalização do modelo.

PONTOS FRACOS

- Tempo de treinamento: O treinamento pode ser demorado, especialmente para grandes conjuntos de dados ou com muitas árvores.
- Tuning de hiperparâmetros: Requer ajustes cuidadosos de parâmetros como a taxa de aprendizado, profundidade da árvore e número de árvores para obter o desempenho ideal.

XGBoost (Extreme Gradient Boosting) é uma implementação avançada do GBM que inclui melhorias significativas.

Utiliza algoritmos otimizados para melhorar a velocidade e a performance.

Proporciona melhor utilização da memória através de técnicas de colunas esparsas.

Suporta paralelização e execução em GPU, acelerando o treinamento.

LightGBM



LightGBM é uma biblioteca de boosting em gradiente que foi projetada para ser rápida e eficiente, especialmente para grandes conjuntos de dados.

Diferente do XGBoost, utiliza um método de 'leaf-wise', que permite segmentar as folhas da árvore, resultando em uma maior eficiência.

Pode ser até 100 vezes mais rápido que outros métodos tradicionais ao lidar com grandes volumes de dados.

Melhor gerenciamento de memória, o que torna o treinamento mais eficiente em termos de recursos.

Ideal para conjuntos de dados que possuem milhões de instâncias, permitindo que análises em larga escala sejam feitas de maneira eficaz.

Tabela Comparativa

Característica	GBM	XGBoost	LightGBM
Velocidade	Mais lento	Rápido	Muito rápido
Escalabilidade	Limitada	Boa	Excelente
Pontos fortes	Simples e flexível	Performance e regularização	Velocidade e grandes bases



Modelos Low Default

Definições



- Exemplos típicos (crédito corporativo, grandes empresas, setores com poucas perdas)
- Características:
 - Alta assimetria
 - Pouquíssimos casos de inadimplência (0,1%, 0,5%, 1%)

Definições



- Problema de desequilíbrio extremo das classes
- Dificuldade na generalização dos modelos
- Modelos que tendem a prever sempre a classe majoritária
- Viés estatístico e métricas enviesadas

Isolation Forest



- Algoritmo não supervisionado para identificar outliers e anomalias.
- Baseado na ideia de que anomalias são mais fáceis de isolar em uma árvore aleatória.
- O default pode ser tratado como um evento raro (anomalia) em relação aos bons pagadores.
- Útil quando há extremo desequilíbrio de classes (quase nenhum default).
- Não requer variável target durante o treinamento — aprende os padrões dos dados majoritários.

Isolation Forest - Funcionamento



- Cria várias árvores que separam os pontos de dados.
- Mede o número de divisões necessárias para isolar cada ponto.
- Pontos que são isolados rapidamente (poucas divisões) são marcados como anômalos.
- Parâmetro $\text{contamination}=0.005$ → define que esperamos cerca de 0,5% de anomalias (ex.: defaults).

Isolation Forest - Saída do Modelo



- Cria uma coluna anomaly_iso:
 - 0 = normal (não default)
 - 1 = anomalia (potencial default)
- Permite transformar um problema não supervisionado em uma pseudo-classificação.

Isolation Forest - Vantagens



- Não precisa de eventos de default para treinar.
- Simples de aplicar e interpretar no contexto de anomalias.
- Rápido, escalável e robusto a grandes volumes de dados.

Isolation Forest - Desvantagens



- Não substitui um modelo supervisionado quando há eventos suficientes.
- A sensibilidade depende da escolha do parâmetro contamination.
- Detecta qualquer tipo de anomalia — não necessariamente apenas defaults.

XGBoost com Peso Balanceado



- Algoritmo de Gradient Boosting baseado em árvores, altamente eficiente e robusto.
- Amplamente utilizado em problemas de classificação, incluindo bases com alto desequilíbrio.
- Lida bem com dados desbalanceados quando configurado corretamente.
- Permite penalizar mais os erros da classe minoritária (defaults).
- Usa o parâmetro `scale_pos_weight` para corrigir o desbalanceamento.

XGBoost com Peso Balanceado- Funcionamento



- Ajusta o peso das amostras da classe minoritária na função de perda.
- Treina um modelo supervisionado, aproveitando as variáveis preditoras.
- Gera a predição na variável `y_pred_xgb`.

XGBoost com Peso Balanceado - Vantagens



- Algoritmo extremamente eficaz para dados estruturados.
- Tem bom desempenho em dados desbalanceados com o ajuste correto.
- Robustez, velocidade e capacidade de capturar interações não lineares.
- Suporta validação cruzada e tuning de hiperparâmetros.

XGBoost com Peso Balanceado- - Desvantagens



- Depende da existência de pelo menos alguns eventos de default.
- Se a base tiver default extremamente raro ($\ll 0,1\%$), pode ser insuficiente isoladamente.
- Pode gerar overfitting se o peso for muito elevado e não houver cuidado no tuning.

SMOTE + Random Forest



- SMOTE (Synthetic Minority Over-sampling Technique):→ Técnica que cria exemplos sintéticos da classe minoritária (default) no espaço das variáveis preditoras.
- Combinado com Random Forest, modelo supervisionado baseado em árvores.
- Útil quando há poucos casos de default.
- Aumenta artificialmente a classe minoritária, equilibrando o dataset.
- Permite treinar modelos supervisionados robustos, mesmo com pouco default.

SMOTE + Random Forest - Funcionamento



- SMOTE: Gera exemplos sintéticos da classe default a partir dos vizinhos mais próximos ($k_neighbors=1$ → agressivo, recomendado quando há pouquíssimos eventos).
- Random Forest: Conjunto de árvores de decisão, treinado no dataset balanceado.
- Predição: Modelo aplicado sobre o dataset original (X) para gerar `y_pred_rf`.

SMOTE + Random Forest - Vantagens



- Resolve diretamente o desbalanceamento pela geração de dados sintéticos.
- Random Forest é robusto, lida bem com ruídos e variáveis correlacionadas.
- Simples de aplicar, com bons resultados práticos.

SMOTE + Random Forest - Desvantagens



- Risco de overfitting: SMOTE pode gerar amostras artificiais pouco realistas se os dados forem escassos ou ruidosos.
- Não funciona bem se os defaults forem extremamente escassos (ex.: 2 ou 3 casos reais).
- Geração de dados sintéticos pode distorcer a distribuição real da variável target.