Máquinas de Gradient Boosting (GBMs) com XGBoost

Os seguintes tópicos são abordados neste tutorial:

* Baixando um conjunto de dados do mundo real de uma competição Kaggle
* Executar engenharia de recursos e preparar o conjunto de dados para treinamento
* Treinando e interpretando um modelo de aumento de gradiente usando XGBoost
* Treinamento com validação cruzada KFold e resultados de agrupamento
* Configurando o modelo de aumento de gradiente e ajustando hiperparâmetros

### **Declaração do problema**

Este tutorial adota uma abordagem prática e focada na codificação. Aprenderemos o aumento de gradiente aplicando-o a um conjunto de dados do mundo real da competição [Rossmann Store Sales](https://jovian.com/outlink?url=https%3A%2F%2Fwww.kaggle.com%2Fc%2Frossmann-store-sales) no Kaggle:

Rossmann opera mais de 3.000 drogarias em 7 países europeus. Atualmente, os gerentes das lojas Rossmann têm a tarefa de prever suas vendas diárias com até seis semanas de antecedência. As vendas nas lojas são influenciadas por muitos fatores, incluindo promoções, concorrência, feriados escolares e estaduais, sazonalidade e localidade.

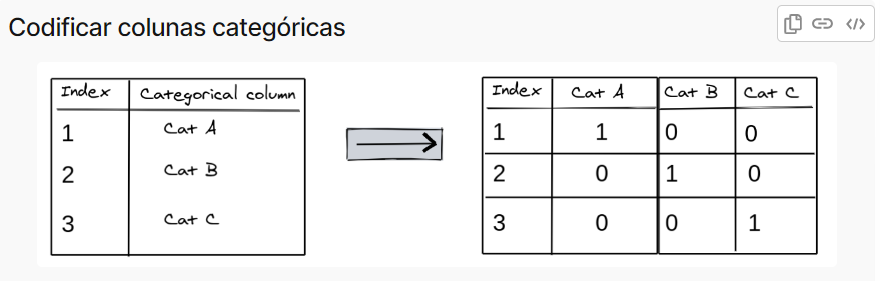
Com milhares de gestores individuais prevendo vendas com base em circunstâncias específicas, a precisão dos resultados pode ser bastante variada. Você recebe dados históricos de vendas de 1.115 lojas Rossmann. A tarefa é prever a coluna "Vendas" para o conjunto de testes. Observe que algumas lojas do conjunto de dados foram temporariamente fechadas para reformas.

Visualize e baixe os dados aqui: [https://www.kaggle.com/c/rossmann-store-sales/data](https://jovian.com/outlink?url=https%3A%2F%2Fwww.kaggle.com%2Fc%2Frossmann-store-sales%2Fdata)

### **Baixando os dados**

Podemos baixar o conjunto de dados do Kaggle diretamente no notebook Jupyter usando a opendatasetsbiblioteca.

Você será solicitado a fornecer suas credenciais do Kaggle para baixar os dados. Siga estas instruções: [http://bit.ly/kaggle-creds](https://jovian.com/outlink?url=http%3A%2F%2Fbit.ly%2Fkaggle-creds)



### **Aumento de gradiente**

Agora estamos prontos para treinar nosso modelo de máquina de aumento de gradiente (GBM). Veja como funciona um modelo GBM:

1. O valor médio da coluna alvo e usa como previsão inicial cada entrada.
2. Os resíduos (diferença) das previsões com os alvos são calculados.
3. Uma árvore de decisão de profundidade limitada é treinada para **prever apenas os resíduos** de cada entrada.
4. As previsões da árvore de decisão são dimensionadas usando um parâmetro chamado taxa de aprendizagem (isso evita overfitting)
5. As previsões dimensionadas para a árvore são adicionadas às previsões anteriores para obter as previsões novas e aprimoradas.
6. As etapas 2 a 5 são repetidas para criar novas árvores de decisão, cada uma delas treinada para prever apenas os resíduos da previsão anterior.

O termo “gradiente” refere-se ao fato de que cada árvore de decisão é treinada com o objetivo de reduzir a perda da iteração anterior (semelhante à descida do gradiente). O termo “boosting” refere-se à técnica geral de treinamento de novos modelos para melhorar os resultados de um modelo existente.

Gostaria de compartilhar uma breve explicação sobre a diferença entre Gradient Boosting Machine (GBM) e Random Forest em machine learning.

**Random Forest:**

O Random Forest é como uma "floresta" de árvores de decisão. Ele treina várias árvores independentes durante o processo de aprendizado, e cada árvore contribui com sua própria opinião para a decisão final. No final, as opiniões são combinadas para produzir um resultado mais robusto e menos propenso a overfitting.

**Gradient Boosting Machine (GBM):**

Por outro lado, o Gradient Boosting Machine constrói árvores sequencialmente, onde cada árvore tenta corrigir os erros das árvores anteriores. É como um aprendizado em equipe, onde cada membro (árvore) foca nos pontos fracos do membro anterior para melhorar o desempenho geral do time.

**Diferenças Chave:**

* **Abordagem de Aprendizado:**
  + Random Forest: Independente e paralela.
  + Gradient Boosting: Sequencial e corretiva.
* **Treinamento:**
  + Random Forest: Cada árvore é treinada de forma independente.
  + Gradient Boosting: Árvores são treinadas para corrigir os erros das árvores anteriores.
* **Robustez e Overfitting:**
  + Random Forest: Menos propenso a overfitting devido à diversidade das árvores.
  + Gradient Boosting: Pode ser mais suscetível ao overfitting se não for ajustado adequadamente.

Espero que essa explicação simples ajude a esclarecer as diferenças fundamentais entre essas duas técnicas valiosas em machine learning. Se tiverem mais perguntas ou insights, fiquem à vontade para comentar!

#MachineLearning #DataScience #RandomForest #GradientBoosting

