Nome completo: Gustavo Boaventura

Primeira pergunta: 48 (PP)

Frequentemente, ocorre uma confusão entre ‘K-NN’ e ‘K-Means’.

Quais as semelhanças e diferenças entre eles? A confusão é válida?

Segunda pergunta: 42 (Oral)

Considere a técnica ‘XGBoost (eXtended Gradient Boost)’.

CITE PELO MENOS DOIS hiperparâmetros que podem ser otimizados quando se usa XGBoost.

https://www.datageeks.com.br/xgboost/

* **XGBoost:** Algoritmo de aprendizado de **máquina da biblioteca do scikit learn**, **baseado na árvore de decisão** (ele é uma **evolução das árvores de decisão**), e utiliza uma estrutura de **Gradient boosting**,
* Ele é considerado um **Ótimo algoritmo para problemas de classificação** assim como **problemas de regressão**, ranqueamento, entre outros.
* ele tambem eh considerado uma evolucao das arvores de decisao
* Tem sido **muito utilizado** em **várias aplicações de ponta na indústria**
* A **escolha** de um algoritmo **depende de diversos fatores**, como **custo computacional**, **prazo de implementação** e acurácia.
* Além disso, **após definir o algoritmo**, a gente precisa utilizar uma **configuração correta**, para um determinado conjunto de dados. Assim se torna de **extrema importância a otimização de hiperparâmetros.**
* Como XGboost se trata de uma árvore de decisão, os dois hiperparâmetros q considero de extrema importância, são **max\_depth e n\_estimators.**
* Estes são importantes para que **não ocorra um problema de overfitting**, que talvez seja o **erro mais comum** quando se aplica modelos de **árvore de decisão**.
* **Overfitting**, eu expliquei no video da atividade anterior, que era sobre árvore de decisão, mas enfim.. eh quando o **algoritmo vicia nos dados de treino** e quando **passa os dados de teste elbu ye erra tudo**.. e a gente não quer isso, a gente quer que ele aprenda o processo, pra na hora que vier os novos dados a gente consiga fazer uma boa previsão.
* Então, o que seria estes hiperparâmetros,
* **max\_depth**: se trata da **profundidade máxima da árvore**. É o número máximo de nós permitido da raiz até a folha mais distante de uma árvore.
* **n\_estimators**: **número de árvores que a gente deseja construir.** ou seja, quanto a maior quantidade de árvores, melhor o desempenho, porém torna o código mais lento. (TEMPO EH $$)?? Sendo que as vezes nao precisa de muitas árvores para chegar na quantidade ótima. Por isso é importante definir bem a quantidade de árvores a serem construídas.
* É isso!