

Exercício 1: Cite cinco diferenças entre o Random Forest e o AdaBoost?

- a. A primeira diferença é que no AdaBoost todas as árvores são do tipo STUMP, ou seja, uma árvore de profundidade tamanho 1 e com duas folhas. No Random Forest as árvores são convencionais, ou seja, podem ter profundidades diferentes;
- b. A segunda diferença é em relação à geração das árvores. No Random Forest elas geradas aleatoriamente. No caso do AdaBoost uma árvore é gerada a partir de uma outra gerada anteriormente. O que de fato acontece é que uma árvore depende da outra para a sua composição;
- c. A terceira diferença é que os resultados das aplicações do algoritmo AdaBoost tem pesos diferentes para cada STUMP. No Random os resultados derivados da aplicação do algoritmo têm o mesmo peso;
- d. A quarta diferença é que no Random Forest os resultados contínuos são usados para se fazer uma média que será o resultado. No AdaBoost os resultados são processados se baseando em combinação ponderada das previsões das árvores – considerando os diferentes pesos;
- e. A quinta diferença é em relação ao tratamento de erros. No Random Forest é menos suscetível a dados ruidosos e outliers, devido à média das previsões de várias árvores. No caso do Adaboost, é mais sensível a outliers, pois dá mais peso aos erros, o que pode levar a *overfitting* se houver ruído excessivo.

Exercício 3: Cite cinco Hyper paramentos importantes no AdaBoost.

- a. **n_estimators**: Número de classificadores fracos (por exemplo, árvores de decisão) a serem criados. Um número maior pode aumentar a performance, mas também pode levar a overfitting.
- b. **learning_rate**: Taxa de aprendizado que controla a contribuição de cada classificador fraco na combinação final. Um valor menor faz com que cada classificador tenha menos impacto, podendo exigir um maior número de estimadores.
- c. **base_estimator**: O modelo base a ser usado como classificador fraco. O padrão é um stump (árvore de decisão de um nível), mas você pode escolher outros modelos, como árvores mais profundas.
- d. **algorithm**: Método utilizado para atualizar os pesos dos classificadores fracos. Os algoritmos comuns são "SAMME" (que pode ser usado com múltiplas classes) e "SAMME.R" (que usa probabilidades para atualizar os pesos).
- e. **max_depth** (quando usando árvores de decisão como base_estimator): Profundidade máxima das árvores de decisão. Limitar a profundidade pode ajudar a evitar overfitting, mantendo os classificadores fracos mais simples.