

Conclusão – Comparação de Modelos com Validação Cruzada

Neste projeto, realizamos a avaliação comparativa de três algoritmos de classificação amplamente utilizados: Decision Tree, Random Forest e XGBoost. O objetivo principal foi analisar o desempenho preditivo de cada modelo por meio de validação cruzada estratificada, utilizando a métrica de acurácia como critério de avaliação.

Primeiramente, organizamos os modelos em um dicionário Python, o que possibilitou iterar facilmente sobre cada algoritmo e aplicar a função `cross_val_score` da biblioteca `scikit-learn`. Essa abordagem permite uma avaliação robusta, já que a validação cruzada com `cv=5` divide os dados em cinco partes, treinando em quatro e testando em uma, repetidamente, e depois calcula a média dos resultados.

Os resultados obtidos foram excelentes para todos os modelos. A Decision Tree apresentou uma acurácia média de aproximadamente **99,65%**, com um desvio padrão extremamente baixo, indicando consistência entre as execuções. O Random Forest também se destacou com acurácia média de **99,57%**, ligeiramente inferior, porém ainda muito precisa. O XGBoost, embora tenha apresentado mensagens de aviso durante a execução, também se mostrou eficaz com acurácia de **99,52%**.

Um ponto importante foi a ocorrência de múltiplos avisos (warnings) relacionados ao uso do parâmetro `use_label_encoder` no `XGBClassifier`. Esses avisos são gerados porque, a partir da versão 1.7 e especialmente na versão 3.0.0 do `xgboost`, esse parâmetro foi descontinuado e não tem mais efeito. A solução definitiva foi **remover completamente o argumento `use_label_encoder=False`** da definição do modelo, mantendo apenas parâmetros relevantes, como `eval_metric='logloss'`.

Esse ajuste elimina os avisos e garante maior compatibilidade com versões futuras da biblioteca. Utilizar `warnings.filterwarnings('ignore')` pode ser útil para ocultar mensagens temporariamente, mas a melhor prática é sempre **resolver a causa raiz do aviso**, como foi feito neste caso.

Em conclusão, a análise demonstrou que todos os modelos são altamente eficazes no conjunto de dados utilizado, com o XGBoost mantendo sua reputação como um dos classificadores mais poderosos, desde que corretamente configurado. A experiência também destacou a importância de acompanhar mudanças nas bibliotecas e manter o código atualizado para garantir desempenho e clareza durante a execução.