Trabalho ML - Extra Trees Classifier

Alunos: Felipe Barroso e Arthur Torquato

1. Introdução

1. Introdução ao Algoritmo

O **ExtraTreesClassifier** foi utilizado para classificar o conjunto de dados de vinhos, separando as instâncias em vinhos de boa e má qualidade. Esse modelo, parte da família de métodos de árvores, seleciona divisões de maneira aleatória, o que o torna menos propenso ao sobreajuste e mais robusto em comparação a métodos tradicionais de Árvores de Decisão. O algoritmo é eficiente tanto para tarefas de classificação quanto de regressão.

Principais Hiperparâmetros

- **n_estimators**: Número de árvores a serem treinadas no modelo.
- **max_depth**: Define a profundidade máxima da árvore para controlar o ajuste excessivo.
- **min_samples_split**: Determina o número mínimo de amostras exigido para dividir um nó.
- **min_samples_leaf**: Número mínimo de amostras exigido para formar uma folha.
- **max_features**: Define o número de características a serem consideradas para cada divisão.

2. Metodologia

Três abordagens principais de otimização de hiperparâmetros foram utilizadas para maximizar o desempenho do **ExtraTreesClassifier**: **RandomizedSearchCV**, **GridSearchCV**, e **BayesSearchCV**.

Técnicas de Otimização Utilizadas

- **Randomized Search**: Primeiramente, uma busca aleatória foi conduzida para explorar amplamente o espaço de hiperparâmetros. Isso permitiu identificar rapidamente as regiões mais promissoras.
- **Grid Search**: Em seguida, uma busca exaustiva com Grid Search foi realizada, focando nos hiperparâmetros mais relevantes identificados na etapa anterior.
- **Bayes Search**: Por fim, utilizamos a Otimização Bayesiana para explorar de forma eficiente os melhores hiperparâmetros, utilizando uma abordagem probabilística para refinar ainda mais os parâmetros.

Hiperparâmetros Testados

- **n_estimators**: Testamos valores de 100 a 1000. O objetivo era encontrar o equilíbrio entre o número de árvores e o tempo de treinamento.
- **max_depth**: Variamos de None até profundidades de 10, 20 e 30, para balancear a capacidade de ajuste e evitar overfitting.
- **min_samples_split** e **min_samples_leaf**: Esses parâmetros foram ajustados para controlar o tamanho das divisões e folhas, reduzindo a variabilidade e garantindo previsões estáveis.

Métricas

- **Acurácia**: Mede a proporção de predições corretas no conjunto de dados.
- **F1-score**: Uma métrica que combina precisão e recall, ideal para conjuntos de dados desbalanceados.
- **Curva ROC e AUC**: Avalia o desempenho do modelo na separação entre classes.

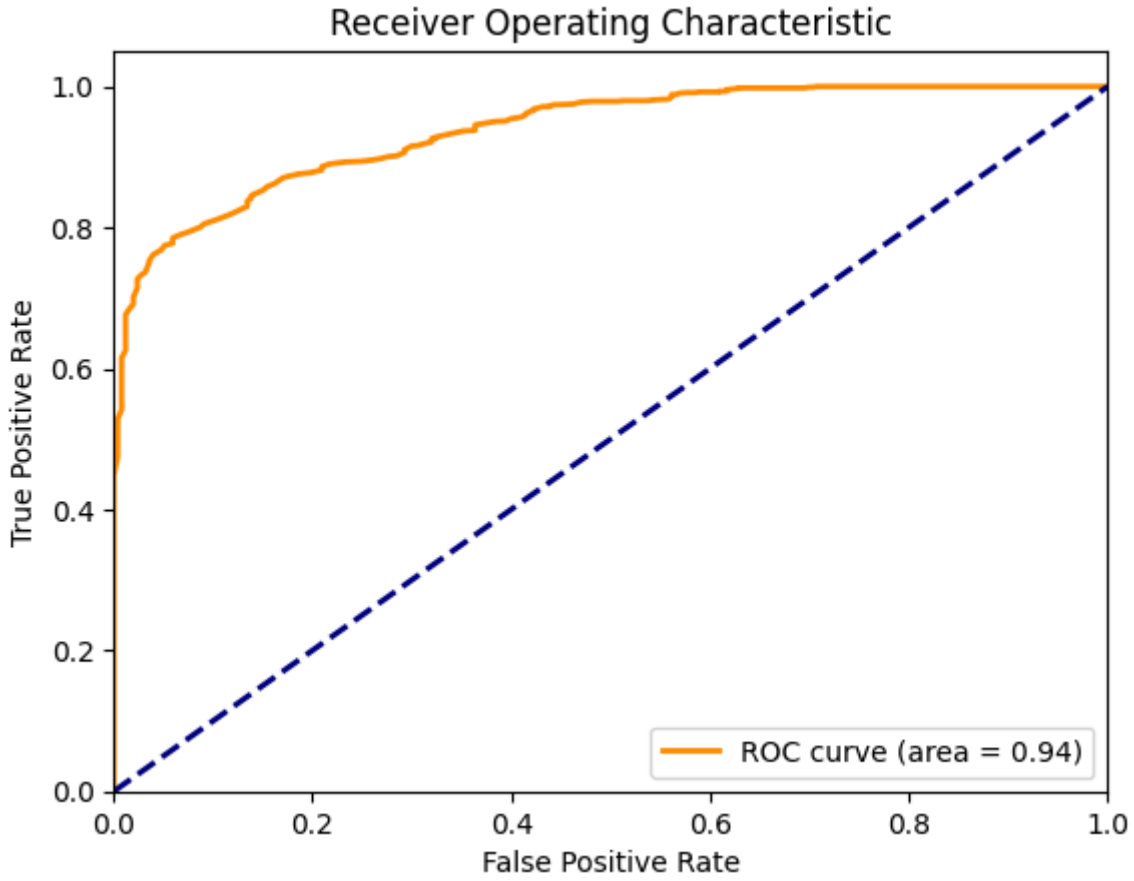
3. Resultados

Resultados para Cada Técnica de Otimização

- **Randomized Search**: Identificou rapidamente parâmetros promissores, resultando em uma acurácia de **82%**. Os melhores hiperparâmetros foram:
    - n_estimators: 200
    - min_samples_split: 2
    - min_samples_leaf: 1
    - max_features: 'log2'
    - max_depth: None
- **Grid Search**: Refinou os resultados, alcançando uma acurácia de **83%**. Os parâmetros encontrados foram:
    - bootstrap: False
    - max_depth: None
    - max_features: 'log2'
    - min_samples_leaf: 1
    - min_samples_split: 2
    - n_estimators: 200
- **Bayes Search**: A Otimização Bayesiana elevou a acurácia para **89%**. Os melhores parâmetros foram:
    - max_depth: None
    - max_features: 'log2'
    - min_samples_leaf: 1
    - min_samples_split: 2
    - n_estimators: 200

Resultados Adicionais:

- **Curva ROC**: A AUC foi de **0.94**, indicando um excelente desempenho na separação das classes.



- **Matriz de Confusão**:

  [[131 122]
  [22 1025]]

  Calculando e exibindo a matriz de confusão. A orientação padrão é a seguinte:

  [0,0]: Verdadeiros Negativos (VN) - Previsões corretamente identificadas como negativas.

  [0,1]: Falsos Positivos (FP) - Previsões incorretamente identificadas como positivas.

  [1,0]: Falsos Negativos (FN) - Previsões incorretamente identificadas como negativas.

  [1,1]: Verdadeiros Positivos (VP) - Previsões corretamente identificadas como positivas.

- **Relatório de Classificação**:

	precision	recall	f1-score	support
0	0.86	0.52	0.65	253
1	0.89	0.98	0.93	1047

accuracy			**0.89**	**1300**
macro avg	**0.87**	**0.75**	**0.79**	**1300**
weighted avg	**0.89**	**0.89**	**0.88**	**1300**

4. Discussão

Melhoria no Processo de Otimização

O uso sequencial de **Randomized Search**, **Grid Search**, e **Bayes Search** provou ser uma estratégia eficiente para otimizar o desempenho do modelo. No entanto, melhorias podem ser feitas com o uso de **Early Stopping** para evitar gastar tempo computacional em treinos desnecessários.

Impacto dos Hiperparâmetros no Desempenho

- **max_depth**: Manter a profundidade ilimitada ajudou a evitar o sobreajuste sem sacrificar a performance.
- **min_samples_split** e **min_samples_leaf**: Valores maiores resultaram em predições mais estáveis e menos suscetíveis a ruídos nos dados.