UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Luis Sávio Gomes Rosa

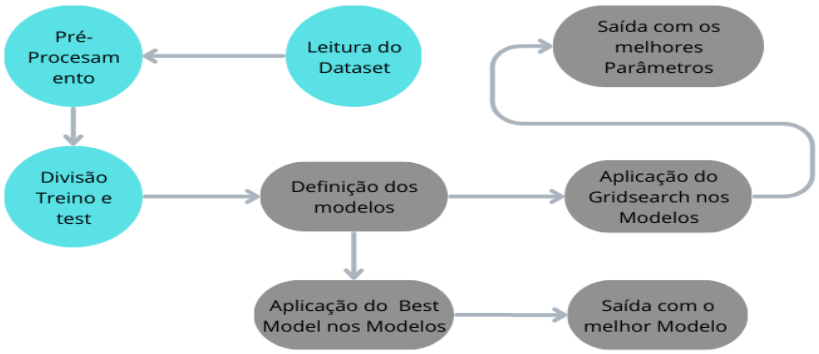
1. Introdução

Neste projeto, implementamos um pipeline automatizado para testar múltiplos modelos de classificação e identificar o modelo com melhor desempenho com base na acurácia. A primeira parte foi uma implementação realizada utilizando o método *GridSearch*. A segunda parte foi a adaptação do GridSearch para retornar o melhor modelo entre todos os testados, levando em consideração a acurácia como métrica principal. Além disso, foram gerados relatórios de classificação detalhados e matrizes de confusão para uma análise mais completa da performance de cada modelo.

Os modelos utilizados foram:

- AdaBoost
- Árvore de Decisão
- Gradient Boosting
- Random Forest
- SVM (Support Vector Machine)

Fluxograma:



2. Sobre o dataset :

Neste conjunto de dados de "Plant Growth Data Classification", a tarefa envolve classificar o marco de crescimento das plantas com base nos fatores ambientais e de gerenciamento fornecidos com base em variáveis como tipo de solo, horas de luz solar, frequência de rega, tipo de fertilizante, temperatura e umidade. Esta ação pode ajudar a compreender como diferentes condições influenciam o crescimento das plantas e pode ser valiosa para otimizar as práticas agrícolas ou a gestão de estufas.

2.1. Pré Processamento:

A priori, a única ação de pré-processamento dos dados foi a transformação dos dados categóricos em dados numéricos. O dataset não possui não valor faltante sua distribuição da target é bem balanceada. O conjunto de treinamento, que contém 80% dos dados e 20% no conjunto de teste.

3. Questão 1

O código trabalha com cinco algoritmos de classificação amplamente utilizados em problemas de aprendizado supervisionado. Cada um desses modelos possui um conjunto de parâmetros (ou hiperparâmetros) que controlam como o algoritmo funciona. Para a otimização dos hiperparâmetros será usado uma técnica chamada GridSearch. Esta técnica realiza uma busca exaustiva por todas as combinações de hiperparâmetros fornecidas para cada modelo de classificação, com o objetivo de identificar quais valores produzem o melhor desempenho.

3.1. Resultados da Questão 1

1. **Decision Tree:**

Melhor parametrização: *{'max_depth': None, 'min_samples_leaf': 10, 'min_samples_split': 20}*

Melhor acurácia: 0.5970

A árvore de decisão apresentou uma acurácia moderada. A configuração ideal incluiu um número mínimo de amostras por folha de 10 e um limite para a divisão de 20 amostras. A ausência de um limite para a profundidade da árvore (max_depth=None) permite que o modelo cresça o suficiente para capturar padrões nos dados, mas com uma quantidade mínima de amostras por folha e uma maior exigência para divisão, o modelo tenta evitar o overfitting. No entanto, essa configuração ainda não foi capaz de superar os métodos de ensemble.

2. **Random Forest:**

- Melhor parametrização: *{'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 100}*
- Melhor acurácia: 0.6295

O Random Forest obteve o melhor desempenho entre os modelos, com uma acurácia de 0.6295. A profundidade máxima de 3 indica que as árvores individuais eram bastante rasas, e isso sugere que o modelo dependia mais da combinação de várias árvores (n_estimators=100) para melhorar a robustez, ao invés de confiar em árvores complexas.

3. **AdaBoost:**

- Melhor parametrização: *{'learning_rate': 1, 'n_estimators': 50}*
- Melhor acurácia: 0.6163

O AdaBoost apresentou uma acurácia de 0.6163, com 50 estimadores e uma taxa de aprendizado de 1. Embora tenha obtido um desempenho competitivo, não superou o Random Forest. Isso pode ocorrer porque o AdaBoost é mais sensível a ruídos no dataset, o que pode impactar negativamente seu desempenho quando comparado a modelos mais robustos como o Random Forest.

4. **Gradient Boosting:**

- Melhor parametrização: *{'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 100}*
- Melhor acurácia: 0.6103

O Gradient Boosting teve um desempenho semelhante ao AdaBoost, com uma acurácia de 0.6103. A baixa taxa de aprendizado (0.01) indica que o modelo ajustou os erros de forma gradual, com o intuito de evitar overfitting. O maior valor de max_depth=5 permitiu que o modelo capturasse padrões mais complexos, mas possivelmente, essa combinação de hiperparâmetros não foi suficiente para superar o Random Forest. O Gradient Boosting tende a ser mais lento e precisar de um ajuste fino mais cuidadoso, o que pode justificar o desempenho próximo do AdaBoost, mas inferior ao Random Forest.

5. **SVM:**

- Melhor parametrização: {'C': 1, 'gamma': 'auto', 'kernel': 'rbf'}
- Melhor acurácia: 0.6105

O SVM com kernel radial obteve uma acurácia de 0.6105, utilizando um valor padrão de C=1 e gamma='auto'. Embora o desempenho tenha sido similar ao Gradient Boosting e AdaBoost, o SVM pode ter tido dificuldades com o volume ou complexidade

dos dados, já que o kernel radial é adequado para capturar padrões não lineares, mas pode não ser a melhor escolha para todos os tipos de dados.

3.2. Discussão sobre os resultados:

O Random Forest foi o modelo com melhor desempenho em termos de acurácia (0.6295). Isso sugere que o uso de modelos de ensemble, combinando previsões de múltiplas árvores de decisão simples, foi mais eficaz do que modelos individuais como a árvore de decisão ou métodos de boosting como AdaBoost e Gradient Boosting. A abordagem do Random Forest de combinar árvores rasas (com max_depth=3) ofereceu uma boa relação entre bias e variância, evitando overfitting e proporcionando previsões mais estáveis.

Modelos como o AdaBoost e Gradient Boosting mostraram resultados próximos, mas sua sensibilidade a ruídos e a necessidade de ajustes mais refinados dos hiperparâmetros podem ter limitado seu desempenho. O SVM, embora eficaz em muitos cenários, pode não ter sido o modelo mais adequado para este conjunto de dados, particularmente devido à sua sensibilidade à escala e à necessidade de um pré-processamento detalhado.

4. Questão 2

Nesta fase, que se assemelha à primeira, o objetivo é identificar o melhor desempenho na classificação. O código retorna melhores parâmetros, o relatório de classificação e a matriz de confusão, permitindo uma análise comparativa detalhada do desempenho de todos os modelos testados.

4.1. Resultados da Questão 2

Os resultados obtidos a partir da avaliação dos modelos de classificação (Árvore de Decisão, Random Forest, AdaBoost, Gradient Boosting e SVM) mostram um cenário onde os modelos de ensemble e a árvore de decisão simples alcançaram desempenhos moderados, enquanto outros modelos, como Gradient Boosting e SVM, tiveram dificuldades em se adaptar ao conjunto de dados. Abaixo, discutimos os principais aspectos de cada um dos modelos testados.

Random Forest:

Acurácia: 0.6410

Classification Report Random Forest

	precision	recall	f1-score	support
0	0.59	0.59	0.59	17
1	0.68	0.68	0.68	22
accuracy			0.64	39
macro avg	0.64	0.64	0.64	39
weighted avg	0.64	0.64	0.64	39

Matriz de Confusão

$$\begin{bmatrix} 10 & 7 \\ 7 & 15 \end{bmatrix}$$

Classificação Geral: O Random Forest obteve a melhor acurácia entre todos os modelos, com um valor de 64.10%. Isso se deve à sua abordagem de ensemble, que combina várias árvores de decisão e reduz o overfitting. A matriz de confusão revela uma distribuição relativamente equilibrada entre as classes 0 e 1, com uma boa precisão e recall para ambas as classes, resultando em um f1-score de 0.68 para a classe majoritária.

Decision Tree:

Acurácia: 0.6154

Classification Report Decisio Tree

	precision	recall	f1-score	support
0	0.56	0.53	0.55	17
1	0.65	0.68	0.67	22
accuracy			0.62	39
macro avg	0.61	0.61	0.61	39
weighted avg	0.61	0.62	0.61	39

Matriz de Confusão

$$\begin{bmatrix} 9 & 8 \\ 7 & 15 \end{bmatrix}$$

Classificação Geral: A Árvore de Decisão foi o segundo melhor modelo, com acurácia de 61.54%. Como uma única árvore, ela mostrou uma leve tendência a overfitting, o que pode ser visto na matriz de confusão. Houve uma quantidade

significativa de previsões incorretas nas duas classes, mas a árvore conseguiu capturar os padrões principais de cada classe.

AdaBoost:

Acurácia: 0.5641

Classification Report AdaBoost

	precision	recall	f1-score	support
0	0.50	0.47	0.48	17
1	0.61	0.64	0.62	22
accuracy			0.56	39
macro avg	0.55	0.55	0.55	39
weighted avg	0.56	0.56	0.56	39

Matriz de Confusão

$$\begin{bmatrix} 8 & 9 \\ 8 & 14 \end{bmatrix}$$

Classificação Geral: O AdaBoost teve um desempenho inferior em comparação com os dois modelos anteriores, com uma acurácia de 56.41%. O modelo teve dificuldades em distinguir corretamente as classes, especialmente a classe 0, onde apresentou uma precisão de 50%. A matriz de confusão indica uma confusão substancial entre as classes. O modelo pode ter sido prejudicado por um dataset com ruído ou classes desbalanceadas, pois o AdaBoost ajusta iterativamente os pesos dos erros de classificação, o que pode levar a um ajuste excessivo a exemplos difíceis ou ruidosos.

Gradient Boosting:

Acurácia: 0.4872

Classification Report Gradient Boosting

	precision	recall	f1-score	support
0	0.41	0.41	0.41	17
1	0.55	0.55	0.55	22
accuracy			0.49	39
macro avg	0.48	0.48	0.48	39
weighted avg	0.49	0.49	0.49	39

Matriz de Confusão

$$\begin{bmatrix} 7 & 10 \\ 10 & 12 \end{bmatrix}$$

SVM

Classificação Geral: O Gradient Boosting foi um dos modelos com pior desempenho, com acurácia de apenas 48.72%. Embora seja um método de boosting como o AdaBoost, ele ajusta gradualmente os erros do modelo. No entanto, a baixa taxa de aprendizado ou a profundidade limitada das árvores pode ter impedido que o modelo capturasse adequadamente os padrões nos dados. Isso é confirmado pela matriz de confusão, onde o modelo apresenta muitas previsões incorretas em ambas as classes, levando a uma baixa precisão e recall.

SVM (Support Vector Machine):

Acurácia: 0.4359

Classification Report SVM

	precision	recall	f1-score	support
0	0.35	0.35	0.35	17
1	0.50	0.50	0.50	22
accuracy			0.44	39
macro avg	0.43	0.43	0.43	39
weighted avg	0.44	0.44	0.44	39

Matriz de Confusão

$$\begin{bmatrix} 6 & 11 \\ 11 & 11 \end{bmatrix}$$

Classificação Geral: O SVM foi o modelo com o pior desempenho, com acurácia de apenas 43.59%. O kernel radial escolhido pode ter sido inadequado para capturar padrões complexos no dataset, ou o modelo pode ter sido impactado por ruído e outliers. A matriz de confusão indica que o modelo teve grande dificuldade em separar corretamente as classes, especialmente a classe 0, onde a precisão foi de apenas 35%.

4.2. Discussão sobre os resultados:

Random Forest foi o modelo de melhor desempenho, alcançando a maior acurácia e mantendo um bom equilíbrio entre as métricas de precisão, recall e f1-score. A Árvore de Decisão teve desempenho próximo, mostrando que ela pode capturar padrões significativos, mas com uma menor capacidade de generalização quando comparada ao Random Forest.

Modelos de boosting, como AdaBoost e Gradient Boosting, não performaram tão bem, indicando que podem ser sensíveis ao dataset utilizado. O AdaBoost, por exemplo,

pode ter enfrentado dificuldades com dados ruidosos, enquanto o Gradient Boosting pode não ter ajustado de forma eficiente os erros.

O SVM foi o modelo de pior desempenho, possivelmente devido à escolha de hiperparâmetros inadequados ou à incapacidade de lidar com a complexidade dos dados.

5. Próximos Passos:

- **Uso de conjuntos de dados maiores:** A análise de conjuntos de dados maiores e mais diversos poderia fornecer insights mais generalizáveis.
- **Incorporação de Variáveis Adicionais:** Incluir variáveis como espécies de plantas, níveis de nutrientes no solo e métodos de controle de pragas poderia oferecer uma compreensão mais abrangente do crescimento das plantas.

6 . Como rodar os algoritmos:

- Para rodar a questão 1, rode o arquivo *grid_search.py*
- Para rodar a questão 2, rode o arquivo *best_model.py*

Ambos estão presentes no diretório do Github

7. Referências

GORTOROZYANNNN. **Plant Growth Data Classification**. Disponível em: <https://www.kaggle.com/datasets/gorororororo23/plant-growth-data-classification/data>. Acesso em: 5 ago. 2024.