Projeto de Aprendizado Supervisionado: Comparando Modelos Clássicos

Objetivo Geral

Aplicar e comparar diferentes algoritmos de aprendizado supervisionado (Redes Neurais Artificiais - RNA, Árvores de Decisão - DT e Random Forest - RF) para resolver problemas reais com dados abertos. Cada grupo deverá desenvolver ao menos **dois modelos diferentes** (entre os três propostos) e apresentar uma análise comparativa dos resultados obtidos.

Contexto

A turma será dividida em **três grupos**, e cada grupo trabalhará em um problema distinto. Todos os problemas são supervisionados, ou seja, envolvem prever uma saída (variável-alvo) com base em entradas conhecidas (atributos).Problemas e Dados

Cada grupo ficará responsável por um dos seguintes problemas:

Grupo	Tema	Tipo de Problema	Dataset	Link
1	Previsão de aprovação escolar	Classificação binária	Student Performance (UCI)	https://archive.ics.uci.edu/dataset/320/student+performance
2	Estimativa de preços de casas	Regressão	Boston Housing (Kaggle)	https://www.kaggle.com/datasets/altavish/boston-housing-dataset
3	Diagnóstico de diabetes	Classificação binária	Pima Indians Diabetes (Kaggle)	https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
4	Previsão de Inadimplência	Classificação binária	Credit Card Default – UCI	https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

Tarefas Obrigatórias

1. **Análise Exploratória de Dados (EDA):**

   Antes de qualquer modelagem, cada grupo deve estudar e entender os dados. Isso inclui:

   - Distribuições das variáveis
   - Correlações
   - Detecção de outliers
   - Visualizações gráficas

2. **Limpeza e preparação dos dados:**

   Tratar valores ausentes, normalizar variáveis se necessário, codificar variáveis categóricas, remover ruído, entre outros.

3. **Seleção e/ou engenharia de características:**

   Avaliar quais variáveis devem ser utilizadas ou transformadas, e se há necessidade de criar novas features a partir das existentes.

4. **Balanceamento de classes (quando necessário):**

   Em problemas de classificação com classes muito desbalanceadas (por exemplo, muitos pacientes saudáveis e poucos com doença), os modelos podem aprender de forma enviesada. Técnicas como undersampling, oversampling ou SMOTE podem ser exploradas.

> Mesmo sem termos discutido essas técnicas em aula, vocês podem pesquisar e aplicar com base em tutoriais confiáveis. O importante é refletir no relatório se o desbalanceamento impactou a performance do modelo.

5. **Construção de modelos:**

   Implementar pelo menos dois dos seguintes algoritmos (Tem que ter RNA):

   - **RNA (Rede Neural Artificial)**
   - **DT (Árvore de Decisão)**
   - **RF (Random Forest)**

6. **Avaliação dos modelos:**

   Utilizar métricas apropriadas para o tipo de problema:

   - Classificação: Accuracy, Precision, Recall, F1-Score, Curva ROC (se possível)
   - Regressão: MAE, RMSE, R^2

7. **Análise Comparativa:**

   Criar uma seção de comparação entre os modelos utilizados:

   - Qual teve melhor desempenho?
   - Qual foi mais interpretável?
   - Qual foi mais rápido de treinar?
   - Que desafios surgiram em cada um?

8. **Relatório Final:**

   Entregar um relatório com linguagem clara, estruturado da seguinte forma:

   - Introdução ao problema
   - Descrição do dataset
   - EDA e preparação dos dados
   - Descrição dos modelos implementados
   - Resultados e comparação entre modelos
   - Conclusões finais com aprendizados do grupo

Observações

- Todos os modelos devem ser implementados em Python (sugestão: usar `scikit-learn` e `keras`).
- O foco principal não é alcançar a maior precisão absoluta, mas sim entender **como o modelo se comporta**, justificar as decisões tomadas e ser capaz de **comparar criticamente os resultados**.
- Trabalhos com bons insights e reflexão crítica serão mais valorizados que apenas altos números de acurácia sem explicação.