Universidade Federal de Mato Grosso do Sul
Sistemas de Apoio à Decisão
Prof. Anderson Bessa da Costa

Atividade EaD (4 horas)
Classificação com Random Forest no Python

Prezado/a Estudante,
Nesta atividade, você irá implementar um modelo de classificação utilizando a linguagem Python e algumas das principais bibliotecas para análise de dados e aprendizado de máquina: pandas, numpy, scikit-learn e matplotlib. Nosso objetivo é analisar um conjunto de dados e construir um classificador baseado no algoritmo Random Forest.

Tarefa

Gere um modelo de classificação para distinguir assinaturas autênticas de falsificadas. Para isso, siga os seguintes passos:

1. **Carregamento dos dados**
   - O conjunto de dados **banknote_authentication.csv** (uma versão csv do arquivo arff que trabalhamos no weka) está disponível no AVA Moodle da disciplina;
   - Utilize a biblioteca *pandas.read_csv* para carregar os dados no Python (considere *import pandas*).

2. **Exploração dos dados**
   - Verifique se há valores faltantes e calcule estatísticas descritivas para cada variável (use *pandas.DataFrame.describe()*).
   - Visualize a distribuição dos atributos com histogramas (use *matplotlib.pyplot.hist()*).

3. **Visualização dos dados**
   - Plote um gráfico de dispersão para verificar a relação entre duas variáveis e a classe (use *matplotlib.pyplot.scatter()*).
   - Análise se alguma variável apresenta boa separabilidade entre as classes.

4. **Análise inicial do problema**
   - Determine o número de instâncias e atributos do conjunto de dados (use *DataFrame.shape*).
   - Verifique se o problema é balanceado analisando a distribuição da classe (use *DataFrame["classe"].value_counts()*).

5. **Construção do modelo de classificação**
   - Divida os dados em treino e teste (use `train_test_split` do *sklearn.model_selection*).
   - Treine um modelo Random Forest (use *RandomForestClassifier* do *sklearn.ensemble*).
   - Utilize validação cruzada com 10 folds (use *cross_val_score* do *sklearn.model_selection*).

6. **Avaliação do modelo**

- Calcule a acurácia e a precisão do modelo (use *accuracy_score* e *precision_score* do *sklearn.metrics*).
- Gere a matriz de confusão e exiba os resultados (use *confusion_matrix*).

7. **Interpretação dos resultados**
   - O problema foi resolvido com sucesso?
   - Algum atributo teve maior importância para o modelo?
   - Os resultados poderiam ser melhorados? Como?

8. **Reanálise com Normalização**
   - Aplique normalização nos atributos (use *StandardScaler* do *sklearn.preprocessing*).
   - Refaça a análise com os dados normalizados e compare os resultados.

Passo a passo

- Revise os conceitos assistindo à videoaula: https://youtu.be/v8fcZHej2wY.
- Crie um notebook no Google Colab e implemente as etapas descritas.
- Submeta sua implementação no AVA Moodle da disciplina.

Esta atividade contabilizará 4 horas-aula de presença.
Bons estudos!