Capacitação IA e IoT
Softex, MCTI
Prof.: Thiago Moura

Projeto Final – Ciclo 1

O objetivo do projeto é consolidar o conhecimento adquirido durante todo o ciclo 1. Aplicar os modelos de aprendizado de máquina aprendidos, configurando seus parâmetros, comparando os resultados obtidos na predição de bases de classificação e regressão.

1) Obter um *dataset* de classificação e outro de regressão, diferentes dos já apresentados durante a capacitação. Os *datasets* podem ser obtidos dos seguintes locais:

- *Datasets* do SkLearn: https://scikit-learn.org/stable/datasets/toy_dataset.html
- *Datasets* do Seaborn: https://github.com/mwaskom/seaborn-data
- *Datasets* do UCI: https://archive.ics.uci.edu/datasets
- *Datasets* do Kaggle: https://www.kaggle.com/datasets
- Qualquer fonte de dados da internet, empresa que trabalha etc.

2) Analisar o *dataset* usando os métodos do Pandas. Comparar as *features* e verificar se há valores nulos, correspondência entre *features*, converter os *labels* para números e *features* categóricas usar *One-Hot Encoder*. Normalizar as *features* na escala 0..1.

3) Apresentar gráfico de barras mostrando a quantidade de instâncias por classe do *dataset* de classificação. Apresentar gráfico de dispersão com linha de regressão, selecionando alguma *feature* do *dataset* de regressão versus a coluna de *target*. Escolher um terceiro gráfico para ser apresentado.

4) Para o *dataset* de classificação:

4.1) Separar os dados de treinamento e teste (80% e 20%);

4.2) Treinar os modelos:

- Duas Árvores de Decisão com variações de parâmetros (critério, profundidade máxima etc);
- Dois KNN com variações de parâmetros (tamanho da vizinhança e métrica de distância);
- Duas MLPs com variações de parâmetros (topologia da rede, número de ciclos, função de ativação);
- Dois *Support Vector Machine - SVM* com variações de parâmetros (**Kernel**, C, gamma);
- Random Forest (escolher os parâmetros das árvores);
- GradienteBoosting, XGBoost ou LightGBM (escolher um dos 3 e configurar os parâmetros como quiser);

4.3) Executar 10 vezes o Train-Test-Split com treinamento e predição dos modelos para cada conjunto de dados randomicamente selecionados;

4.4) Apresentar a média das 10 execuções dos resultados de acurácia de todos os modelos. Apresentar os resultados com 3 casas decimais, por exemplo: 93,123%;

5) Para o *dataset* de regressão:

5.1) Separar os dados de treinamento e teste (80% e 20%);

5.2) Treinar os modelos:

- Duas Árvores de Decisão para regressão com variações de parâmetros (critério, profundidade máxima etc);
- Dois KNN para regressão com variações de parâmetros (tamanho da vizinhança e métrica de distância);
- Duas MLPs para regressão com variações de parâmetros (topologia da rede, número de ciclos, função de ativação);
- Dois *Support Vector Regressor - SVR* com variações de parâmetros (**Kernel**, C, gamma);
- Random Forest para regressão (escolher os parâmetros das árvores);
- GradienteBoosting, XGBoost ou LightGBM (escolher um dos 3 e configurar os parâmetros como quiser);

5.3) Executar 10 vezes o Train-Test-Split com treinamento e predição dos modelos para cada conjunto de dados randomicamente selecionados;

5.4) Apresentar a média das 10 execuções dos resultados de MSE, RMSE e MAE de todos os modelos;

DATA DE ENTREGA: 16 de Novembro de 2023

HORA: até 20:00h

Observação: PROJETO INDIVIDUAL!