Exercício

Vamos criar uma árvore de decisão para classificar contas de cartão de crédito quanto à inadimplência.

Crie um novo notebook no Google.Colab, carregue os seguintes pacotes:

- **numpy** → para cálculos em geral
- **pandas** → preparação dos dados
- **matplotlib.pyplot** → plotagem
- **graphviz** → visualização da árvore
- **train_test_split** do **sklearn.model_selection**
- **tree** do **sklearn**
- **metrics** do **sklearn**

} **scikit-learn** : partição, árvores e avaliação

Use o pandas para carregar o dataset Cartões_dados_limpos.csv

Inspecione os nomes das colunas do DataFrame criado usando .columns.tolist( )
Aliás, observe o aspecto geral dos dados.

Exercício

Vamos fazer uma lista das colunas que não serão usadas neste caso: ID, SEX, PAY_2 até PAY_6, EDUCATION_CAT, graduate school, high school, none, others e university. Use esta lista para gerar outra, contendo apenas as demais: usaremos a segunda no treinamento do modelo.

segunda_lista = [<se for coluna que não está na primeira>] (ou seja, *list comprehension)*

Vamos agora ao treinamento do modelo. Divida o DataFrame em conjunto de treinamento e conjunto de testes, usando *train_test_split( )*. Escolha a porcentagem de dados deixada para teste (usualmente 20% ou 30 %). Lembrem-se de usar a lista das colunas separando a última (***default payment next month***) que armazena a variável de resposta (a classe). O método *values( )* dá acesso aos valores.

Em seguida instancia-se o modelo usando-se a classe *DecisionTreeClassifier*. Podemos começar com um valor de *max_depth = 2*.

Tendo o objeto do modelo, vamos treiná-lo usando **<variável do modelo>.fit (<X de treino, Y de treino>)**

Exercício

Neste ponto temos um modelo treinado, ou seja, foi construída uma árvore de decisão baseada nos dados do dataset.

Para visualizá-la vamos usar o **graphviz**:

dados_diag = tree.export_graphviz (<variável do modelo>, out_file=None, filled=True, rounded=True, feature_names=<nome da segunda lista>[:-1], proportion=True, class_names=['Em dia', 'Inadimplente'])

Examinem na documentação do pacote o significado de cada parâmetro.

A visualização foi construída e não foi salva em arquivo. Para exibi-la usa-se o método .Source( ) :

```
grafico = graphviz.Source(dados_diag)
grafico
```

Examinem a representação da árvore. Qual foi o índice de ganho de informação usado para os pontos de corte? Como ficaria a árvore com outro índice? Como poderíamos alterar este índice? Experimentem treinar modelos com parâmetros diferentes, começando pela profundidade máxima.

Exercício

Examinem a representação da árvore. Qual foi o índice de ganho de informação usado para os pontos de corte? Como ficaria a árvore com outro índice? Como poderíamos alterar este índice? Experimentem treinar modelos com parâmetros diferentes, começando pela profundidade máxima.

Nas próximas aulas abordaremos alternativas de treinamento para melhorar o modelo. Por enquanto, avaliem o desempenho do que implementarem usando o método predict( ) e passando seu resultado para metrics.accuracy_score(<lista de valores esperados no teste, lista predita>).

Posteriormente iremos analisar as possíveis métricas de avaliação de modelos.