Lista de Exercícios 06

(prazo final para entrega: 14/04/2019 - domingo)

1) Implemente as 2 classes a seguir e adicione à sua biblioteca de pre-processamento em Python no arquivo **transform.py** :
   a) **Normalize**
   b) **Standardize**

   Ambas devem ter os métodos **fit(X)** e **transform(X)**.

   - fit - percorre a matriz de entrada e obtém os valores necessários aos cálculos específicos de cada classe:
     i. Normalize obtém min e max.
     ii. Standardize obtém mean e std.

   - transform - altera os valores da matriz de entrada de acordo com a operação realizada (normalizadação ou estandardização)

2) Implemente uma função **split_stratified_train_test** para dividir os dados de treino e teste no arquivo **resample.py** a ser incluído em sua biblioteca Python. A função deve permitir:
   a) randomização dos dados.
   b) escolha de percentual dos dados usados para treino (o complemento desse percentual representa os dados usados para teste).
   c) estratificação dos dados (ver link).

   Assinatura: **split_stratified_train_test(y, perc_train, seed)**
   - perc_train - percentual dos dados usados para treino.
   - seed - semente para geração de números randômicos.
   - y - labels de cada observação contendo a classe como valor.
   - saída (output): array estratificado com os índices do dados de treino e array estratificado com os índices dos dados de teste.
     i. Exemplo de entrada: split_stratified_train_test(0.7, 42, y)

     Saída - 2 arrays com índices dos dados de treino (idx_train) e de teste (idx_test), respectivamente.

3) Use as implementações acima para tratar o dataset de renda de americanos dos Estados Unidos, que tem como rótulo se a pessoa ganha ou não mais de 50.000 dólares por ano.
   a) O dataset contém dados categóricos e valores faltando.
   b) Faça o melhor pre-processamento possível para tornar tal dataset adequado para uso nos algoritmos de aprendizagem de máquina para classificação. Use 70% dos dados para treino e outros 30% para teste.
   c) Faça atribuição da média da coluna para valores faltantes, caso o percentual de valores faltantes da coluna não seja muito grande.

d) Transforme categorias usadas nas features e label em números. Categorias que não possuem uma ordem implícita devem ser transformadas em features binárias.
e) Compare os resultados obtidos entre os algoritmos acima usando a métrica accuracy (implemente-a no arquivo **metrics.py** de sua bliblioteca Python).