MAB225 – Computação II – Revisão de Numpy

Para os exercícios abaixo, considere os seguintes imports:

```
import numpy as np
from matplotlib import pyplot
import matplotlib.image as mpimg
import matplotlib.pylab as plt
import numpy.matlib as npmat
import scipy.optimize as spopt
```

1. Crie um código que, através dos passos descritos abaixo, adiciona uma marca d'água com a logo do DCC a uma imagem. Tanto a logo quanto a imagem foram carregadas:

   ```
   imagem = mpimg.imread("Lennon.jpg")
   logo = mpimg.imread("logo.jpg")
   ```

   a. Desejamos reduzir o tamanho original da logo. Uma forma comum de reduzir a resolução de uma imagem consiste em aplicar um filtro passa-baixas a ela e, em seguida, sub-amostrar a imagem resultante. Utilizando o numpy, esse procedimento pode ser realizado da seguinte forma:
      i. Crie uma cópia da variável logo utilizando a função copy e transforme o tipo de dados do array copiado para "float64". Faça o mesmo para a imagem.
      ii. Calcule a média de cada bloco de 4x4 pixels do array com elementos tipo "float64". Não há intersecção entre os elementos. Exemplo:

      iii. Crie uma nova variável amostrando um pixel de cada bloco.
   b. Replique a logo reduzida de forma a criar uma imagem RGB com 2 X 3 logos.
   c. Multiplique cada pixel da logo repetida por 0.5 e some o resultado aos pixels do canto inferior direito da imagem tipo "float64". A soma deve ser feita de forma que a logo repetida fique exatamente no canto inferior direito, colada à borda. Para tal, utilize shape para descobrir o número de linhas e número de colunas da logo e da imagem.
   d. Faça com que todos os pixels da imagem cujo resultado da soma foi maior que 255 fique igual a 255. Esse passo encerra a criação da marca d'agua.
   e. Para terminar a aplicação do numpy em processamento de imagens, transforme a imagem resultante para escala de cinza. Para transformar uma imagem colorida representada por três canais de cores – uma matriz com a quantidade de vermelho de cara pixel, outra com a quantidade de verde e outra com a quantidade de azul – realizamos, para cada pixel, um somatório ponderado do valor vermelho, verde e azul. O valor do pixel cinza é dado por:

      $$pCinza_{x,y} = 0.2989 \cdot pVermelho_{x,y} + 0.5870 \cdot pVerde_{x,y} + 0.1140 \cdot pAzul_{x,y}$$

      Escreva um código em Python que transforme cada pixel no seu equivalente cinza de acordo com a equação acima. A imagem resultante deve ter três dimensões onde o valor do pixel em uma determinada posição de cada dimensão é igual e igual ao valor dado pela equação.

2. Crie um código para fazer a análise estatística do arquivo "kc_house_data.csv", que contém uma tabela com informações do valor de casas nos Estados Unidos.
   a. Carregue o arquivo lendo somente as colunas 2, 3 e 5 (referentes a preço, quantidade de quartos e área da casa). Utilizando o pyplot, plote a área no eixo x e o preço no eixo y para ver a relação entre os pontos (não é necessário decorar como utilizar o pyplot para a prova, mas o loadtxt é importante).

b. Imprima a matriz de correlação entre área e preço.
c. Utilizando a função curve_fit do módulo spopt, calcule a reta que melhor se ajusta ao conjunto de pontos da área versus preço. Escreva o código necessário para utilizar os valores retornados pelo curve_fit de forma a calcular o valor dos preços estimados dado um vetor de áreas. Plote o resultado.
d. Como existem poucos pontos acima de 8000 de área, crie arrays do numpy com somente os valores de área menores que 8000 e os valores equivalentes dos preços das casas para essas casas com área menor que 8000. Realize um novo ajuste de reta (curve_fit) para esse novo conjunto de pontos. Plote o resultado.
e. Utilizando o pyplot, plote a área no eixo x e a quantidade de quartos no eixo y para ver a relação entre os pontos – observe que o log pode ser uma boa curva de aproximação.
f. Utilizando a função curve_fit do módulo spopt, calcule a função a*np.log(x)+b que melhor se ajusta ao conjunto de pontos da área versus versus quantidade de pontos. np.log() é uma função do numpy. Escreva o código necessário para utilizar os valores retornados pelo curve_fit de forma a calcular a quantidade de quartos estimada dado um vetor de áreas. O que acontece se o vetor de áreas não estiver ordenado? Plote o resultado.
g. Repita a questão acimara para a função a*np.log(x+c)+b. O que muda no código?