UNIVERSIDADE ESTADUAL DE LONDRINA
Especialização em Machine Learning e Big Data
Fundamentos Matemáticos e Computacionais de Machine Learning

Profa. Dra. Juliana Felix
jufelix16@uel.br

Regressão Linear e Logística

Instruções: Você pode utilizar o ambiente de desenvolvimento de sua preferência. No entanto, sua solução deve ser enviada em um único arquivo python notebook (ipynb) em que cada célula corresponde a um item ou exercício. Identifique cada questão adequadamente.

Data de entrega: Até 23h59 do dia 22/05/2023

Regressão Linear

1. Considerando os valores x e y fornecidos, tente encontrar, **manualmente**, pela atribuição de valores, uma reta que se ajuste aos dados abaixo:

```
x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])
```

Faça isso utilizando:

a. Apenas θ_0 (ou, equivalentemente, b)
b. Apenas θ_1 (ou, equivalentemente, m)
c. Atribuindo valores para θ_0 e θ_1 (b e m, respectivamente)

Calcule o respectivo MSE, plote as retas e apresente o MSE encontrado em todos os casos.

Código de suporte

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])

plt.plot(x, y, 'o', color='black');
plt.xlim(0, 2500)
plt.ylim(0, 400)
plt.xlabel('Área em $m^2$')
plt.ylabel('Preço em 1000\'s R$')
plt.title('Preço estimado de um lote')
plt.grid()
plt.show()
```

Profa. Dra. Juliana Felix
jufelix16@uel.br

2. Considerando os valores x e y fornecidos, crie um algoritmo para encontrar, **automaticamente**, uma reta que melhor se ajuste aos dados abaixo:

```
x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])
```

Faça isso utilizando:

a. Apenas θ_0 (ou, equivalentemente, b)
b. Apenas θ_1 (ou, equivalentemente, m)

Para cada caso, plote a reta e apresente o MSE encontrado como melhor solução.

Dica: Siga o fluxo de treinamento mostrado em aula, repetindo o algoritmo até que a condição de parada estabelecida por você seja satisfeita.

3. Considerando os valores x e y fornecidos, **implemente** o algoritmo do **gradiente descendente** para encontrar os valores de θ_0 e θ_1 que formam a reta que melhor se ajusta aos dados abaixo. Plote a reta, e apresente os valores de θ_0 e θ_1 encontrados como melhor solução.

```
x = np.array([480, 510, 520, 850, 960, 1200, 1400, 1650, 1700, 1920, 2350])
y = np.array([98, 110, 200, 210, 280, 265, 300, 287, 325, 300, 290])
```

Regressão Logística

4. Considerando os valores x1, x2 e y fornecidos, **implemente** o algoritmo da **regressão logística** para encontrar uma reta que separa (classifica) os dados baseado em y:

```
x1 = np.array([0, 1, 0, 1])
x2 = np.array([0, 0, 1, 1])
 y = np.array([0, 0, 0, 1])
```

Plote cada classe com uma cor característica e plote a reta encontrada, apresentando os valores de θ_0, θ_1 e θ_2 encontrados como melhor solução.

Dica: O plote esperado deve relacionar a feature 1 (x1) com a feature 2 (x2).

Profa. Dra. Juliana Felix
jufelix16@uel.br

5. Considerando os valores x1, x2 e y fornecidos, **implemente** o algoritmo da **regressão logística** para encontrar uma reta que separa (classifica) os dados baseado em y:

```
x1 = np.array([0, 1, 0, 1])
x2 = np.array([0, 0, 1, 1])
 y = np.array([0, 1, 1, 1])
```

Plote cada classe com uma cor característica e plote a reta encontrada, apresentando os valores de θ_0, θ_1 e θ_2 encontrados como melhor solução.

Dica: O plote esperado deve relacionar a feature 1 (x1) com a feature 2 (x2).