PPGEE2249 - Aprendizado de Máquina

Pedro Ahim - 180208042

① a) "+1" $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-2)^2} = p(x|C_{+1})$

"-1" $f(x) = \begin{cases} 1/4 & -2 < x < 2 \\ 0, & \text{otherwise} \end{cases} = p(x|C_{-1})$

Bayes Rule

$$P(C|x) = \frac{P(C)\,p(x|C)}{p(x)}$$

Decision

choose C_i if $P(C_i|x) = \max_k P(C_k|x) = \max_k P(C_k) \cdot p(x|C_k)$

"+1" if $0{,}6 \cdot \overbrace{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-2)^2}}^{p(x|C_{+1})} > 0{,}4 \cdot \frac{1}{4}$

"-1" if $0{,}4 \cdot \frac{1}{4} > 0{,}6 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-2)^2}$

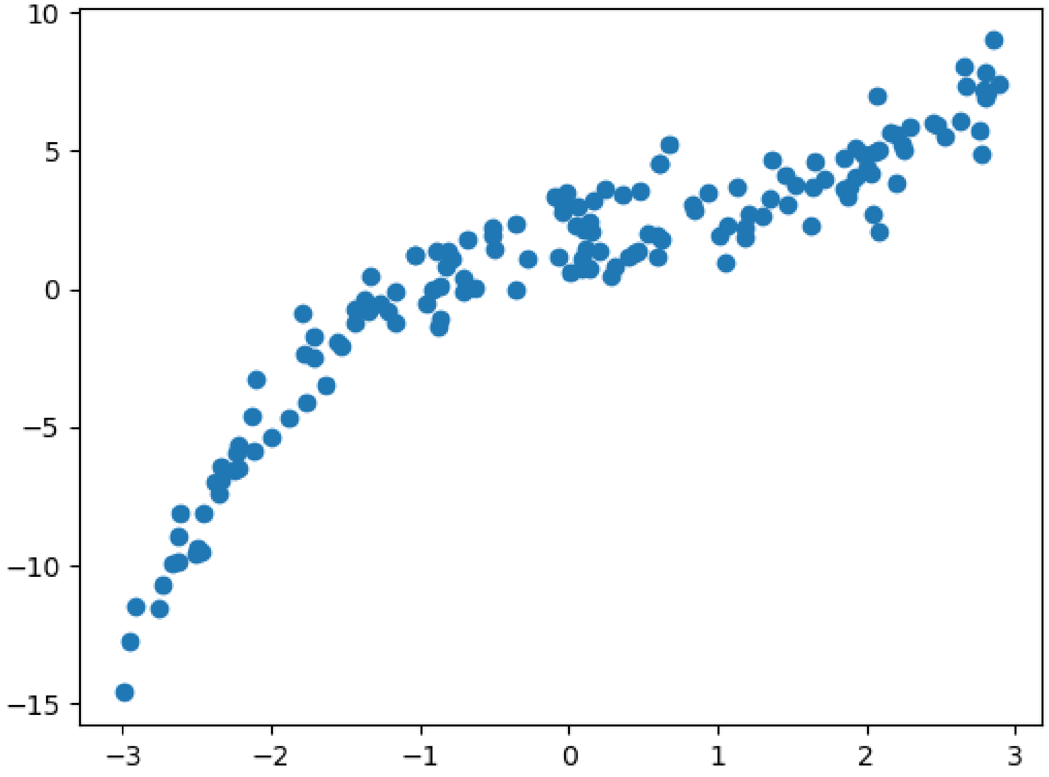
b) Decision rule:

$$p(x|C_{+1}) > \frac{1}{6}$$

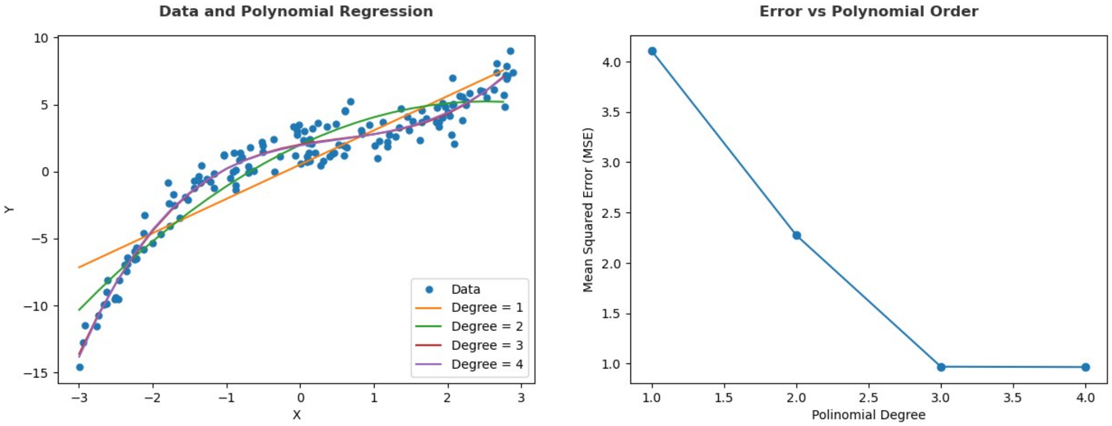
$$\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-2)^2} > \frac{1}{6}$$

```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt

N = 150
x = 6*np.random.rand(N,1) - 3
y = 0.3 * x**3 - 0.5 * x**2 + x + 2 + np.random.randn(N,1)

plt.plot(x,y, 'o')
```

```
[<matplotlib.lines.Line2D at 0x7f20013110f0>]
```



```
def polinomial_regression(degree, x, y):
    l = []
    a_arr = []
    y_r = []

    #A
    #Lines
    for i in range(degree + 1):
        #Rows
        l = []
        for j in range(degree + 1):
            if i == 0 and j == 0:
                l.append(N)
            else:
                l.append(sum(x**(i+j))[0])
        a_arr.append(l)

    a = np.matrix(a_arr)
    a_1 = np.matrix.getI(a)

    #Y
    for i in range(degree + 1):
        y_r.append([sum(y*x**i)[0]])

    w_matrix = np.matmul(a_1, y_r)

    return w_matrix

def calculate_poli(x_data, w):
    x = np.arange(min(x_data),(max(x_data)),0.25)
    y = 0
    for i in range(len(w)):
        y += w[i][0] * x**i

    return x, y

def mse(degree, x, y, w):
    n = len(x)

    y_regression = 0
    for i in range(len(w)):
        y_regression += w[i][0] * x**i

    s = np.sum((y - y_regression)**2)

    return s/n

degree = 5

fig, (ax1, ax2) = plt.subplots(1, 2)

fig.set_figwidth(15)
fig.set_figheight(5)

ax1.plot(x,y, 'o', markersize=5, label='Data')

error = []
error_x = []

for i in range(1,degree):
    pol = polinomial_regression(i, x, y)
    w_pol = np.array(pol)

    x_pol , y_pol = calculate_poli(x, w_pol)

    error.append(mse(i, x, y, w_pol))
    error_x.append(i)

    s = 'Degree = '+str(i)
    ax1.plot(x_pol, y_pol, label=s)


ax2.plot(error_x, error, '-o')

ax1.set(xlabel='X', ylabel='Y')
ax1.set_title('Data and Polynomial Regression',
          pad=15, color='#333333', weight='bold')
ax1.legend(loc='lower right')

ax2.set(xlabel='Polinomial Degree', ylabel='Mean Squared Error (MSE)')
ax2.set_title('Error vs Polynomial Order',
          pad=15, color='#333333', weight='bold');
```


Data and Polynomial Regression
Y
X
Data
Degree = 1
Degree = 2
Degree = 3
Degree = 4
Error vs Polynomial Order
Mean Squared Error (MSE)
Polinomial Degree