STQQSSD.	
PPGEE2243 - Aprendigada de Magnina	6 50 5 5 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
Pedro Ahim - 180208042	
	24,000
$\frac{\langle 2\pi \rangle}{\sqrt{2\pi}}$	Asserting to the second
	to the company of these
"-1" $f(x) \leq y_4 - 2 < x < 2 = p(x C-1)$ 0, otherwise	A A A A A A A A A A A A A A A A A A A
Lo, otherwia	1
	The state of the s
Bayes Rule $P(C x) = P(C) \rho(x C)$	
$P(C x) = \frac{P(C x)}{P(x)}$	The Secretary of Association
Decision	The second secon
choose Ci if P(Cilx) = max P(Ck x) = max P(Ck) p(x)	-k)
	Mar physics
(x(C+1)	
$\frac{1}{1+1} \text{ if } 0,6 - \frac{1}{1} \text{ e}^{\frac{1}{2}(x-2)^2} > 0,41 \cdot \frac{1}{4}$	
하는데 마른데 마른 그런 경우는 보고 살아왔다. [18] 전화 사람들은 아이에게 되었다면 하는데	And production of the second
$\frac{-1}{4}$ if $0,4 \cdot \frac{1}{4} > 0,6 \cdot \frac{1}{2\pi}$	hand the transfer of the section of
b) Descin Tole	
b(×1C→1)>1	
그들은 마음이 얼마나 아내는	
$\frac{1-e^{-\frac{1}{2}(x-2)^2}>1}{\sqrt{2\pi}}$	erektina pieri a peraktivalny. Stepania ori orași de la VIII pa
Sold a ment of the state of the	P)
	The property of the party of th
	A STATE OF THE STA

```
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>]
    10
     5
     0
   -5
  -10
  -15
        -3
                 -2
                          -1
                                                    2
                                            1
                                                             3
def polinomial_regression(degree, x, y):
    [] = []
    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)
    V = 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
                                                    Error vs Polynomial Order
                                         4.0
                                         3.5
                                         3.0
                                        3.0
2.5
2.5
                                         2.0
                               Degree = 1
                                         1.5
                               Degree = 2
                                         1.0
                               Degree = 4
                                                       Polinomial Degree
```