Nashirudin Baqiy
24060119130045
Lab A1

BAB I
PENDAHULUAN

1.1 Tujuan

1. Dapat menentukan penyelesaian model regresi secara Numerik.
2. Mencari besarnya kesalahan dari suatu perhitungan solusi model regresi secara numerik.

1.2 Rumusan Permasalahan

Cocokkan polinomial orde kedua ke data berikut

x_i	0.05	0.15	0.46	0.70	0.82	1.17
Y_i	0.956	0.832	0.571	0.378	0.306	0.104

Persamaan model polinomial orde dua $y = a_1 + a_2x + a_3x^2$

Misalkan $f_1(x_i) = 1, f_2(x_i) = x_i$ and $f_3(t_i) = x_i^2$, **a**= $[a_1\ a_2\ a_3]^T, n = 3, N = 6$

Tentukan solusi model regresi polinomial orde dua tersebut!

Nashirudin Baqiy
24060119130045
Lab A1

BAB II
DASAR TEORI

2.1 Pendahuluan Teori

Dalam penerapan model regresi, satu tujuan adalah untuk mendapatkan persamaan y=f(x) yang paling menggambarkan n titik-titik data respon (x1 ,y1), (x2 ,y2), ..., (xn , yn). Akibatnya, kita dihadapkan dengan jawaban dua pertanyaan dasar.

1. Apakah model y = f (x) menggambarkan data secara memadai, yaitu apakah ada kecocokan yang memadai?

2. Seberapa baik model memprediksi variabel respons (prediktabilitas)?

Lihatlah dua persamaan yang diberikan di bawah ini.

$$S_r = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\ \right)^2 = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2$$

$$S_t = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

dimana Sr adalah jumlahan kuadrat dari residu (residu adalah perbedaan antara nilai observasi, yi dan nilai prediksi,$\widehat{y_l}$), dan St adalah jumlahan kuadrat dari perbedaan antara nilai obervasi dan nilai rata-rata.

2.2 Metode Regresi

Diberikan N titik data, $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, kemudian mencari persamaan polinomial, sebagai

$$P_n(x) = a_1 + a_2 x + \cdots + a_n x^n$$

Atau dapat ditulis sebagai

$$P_n(x) = \sum_{i=1}^{n} a_i x^i$$

Algoritma Regresi :

1. Menghitung sum(x), sum(x^2), sum(x^3), sum(x^4)

2. Membuat matrix $A = \begin{bmatrix} n & sum(x) & sum(x^2) \\ sum(x) & sum(x^2) & sum(x^3) \\ sum(x^2) & sum(x^3) & sum(x^4) \end{bmatrix}$
3. Menghitung sum(y), sum(xy), sum(x^2y)
4. Membuat matrix $B = \begin{bmatrix} sum(y) \\ sum(xy) \\ sum(x^2y) \end{bmatrix}$
5. Invers matrix A
6. Kalikan invers matrix A dengan matrix B
7. Hasilnya $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = Hasil\ kali\ matrix$

2.3 Metode Eliminasi Gauss

Diberikan N titik data, $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, kemudian mencari persamaan polinomial, sebagai

$$P_n(x) = a_1 + a_2 x + \cdots + a_n x^{n-1}$$

Atau dapat ditulis sebagai

$$P_n(x) = \sum_{i=1}^{n} a_i x^i$$

Algoritma Eliminasi Gauss:

1. Menghitung sum(x), sum(x^2), sum(x^3), sum(x^4), sum(y), sum(xy), sum(x^2y)
2. Membuat matrix $A = \begin{bmatrix} n & sum(x) & sum(x^2) & sum(y) \\ sum(x) & sum(x^2) & sum(x^3) & sum(xy) \\ sum(x^2) & sum(x^3) & sum(x^4) & sum(x^2y) \end{bmatrix}$
3. Untuk m=0 sampai n:
   3.1 Untuk r=m sampai n
      3.1.1 lead = A[r][m]
      3.1.2 untuk k=0 sampai n+1, A[r][k] = A[r][k]/lead
   3.2 Untuk r=n-1 sampai 0:
      3.2.1 untuk k=0 sampai n+1, A[r][k] = A[r][k] - A[r-1][k]
4. Untuk i=n-1 sampai 0:
   4.1 x[i] = A[i][n]/A[i][i]
   4.2 untuk k=i-1 sampai 0, A[k][n] = A[k][n] - A[k][i]*x[i]
5. Kembalikan nilai x

BAB III
PEMBAHASAN

3.1 Penyelesaian dengan Metode Regresi

3.1.1 Source Code

```
# Metode Regresi
# 24060119130045 - Nashirudin Baqiy

import numpy as np
import pandas as pd

#data pada tabel
x = [0.05, 0.15, 0.46, 0.70, 0.82, 1.17]
y = [0.956, 0.832, 0.571, 0.378, 0.306, 0.104]

def x2(x):
    return x*x
def x3(x):
    return x*x*x
def x4(x):
    return x*x*x*x
def regresi(x, y):
    #Panjang/jumlah data point
    n = len(x)
    #Inisiasi tabel
    tbl = [[None for x in range(7)] for x in range(n+1)]
    #Sum x
    sx = 0
    for i in range(n):
        sx = sx + x[i]
        tbl[i][0] = x[i]
    tbl[n][0] = sx
    #Sum y
    sy = 0
    for i in range(n):
        sy = sy + y[i]
        tbl[i][1] = y[i]
    tbl[n][1] = sy
    #Sum x2
    sx2 = 0
    for i in range(n):
        sx2 = sx2 + x2(x[i])
        tbl[i][2] = x2(x[i])
    tbl[n][2] = sx2
    #Sum x3
    sx3 = 0
    for i in range(n):
        sx3 = sx3 + x3(x[i])
        tbl[i][3] = x3(x[i])
    tbl[n][3] = sx3
```

```
    #Sum x4
    sx4 = 0
    for i in range(n):
        sx4 = sx4 + x4(x[i])
        tbl[i][4] = x4(x[i])
    tbl[n][4] = sx4
    #Sum x.y
    sxy = 0
    for i in range(n):
        sxy = sxy + x[i]*y[i]
        tbl[i][5] = x[i]*y[i]
    tbl[n][5] = sxy
    #Sum x2y
    sx2y = 0
    for i in range(n):
        sx2y = sx2y + x2(x[i])*y[i]
        tbl[i][6] = x2(x[i])*y[i]
    tbl[n][6] = sx2y

    tbl_table = pd.DataFrame(tbl,
columns=['x','y','x2','x3','x4','xy','x2y'])
    print(tbl_table)
    print()
    print('row 6 adalah sum')

    a = np.linalg.inv([ [n, sx, sx2], [sx, sx2, sx3], [sx2, sx3, sx4]
])
    b = [sy, sxy, sx2y]
    hasil = np.matmul(a, b)
    print()
    print('Didapatkan: %f + %fx + %fx^2' % (hasil[0], hasil[1],
hasil[2]))

regresi(x,y)
```

3.1.2 Hasil Command Prompt

```
(base) F:\TUGAS\Praktikum\Metnum\Tugas3\py>regresi.py
      x      y      x2        x3        x4       xy       x2y
0  0.05  0.956  0.0025  0.000125  0.000006  0.04780  0.002390
1  0.15  0.832  0.0225  0.003375  0.000506  0.12480  0.018720
2  0.46  0.571  0.2116  0.097336  0.044775  0.26266  0.120824
3  0.70  0.378  0.4900  0.343000  0.240100  0.26460  0.185220
4  0.82  0.306  0.6724  0.551368  0.452122  0.25092  0.205754
5  1.17  0.104  1.3689  1.601613  1.873887  0.12168  0.142366
6  3.35  3.147  2.7679  2.596817  2.611396  1.07246  0.675274

row 6 adalah sum

Didapatkan: 0.998304 + -1.053896x + 0.248467x^2
```

3.1.3 Excel

i	xi	yi	xi^2	xi^3	xi^4	xi.yi	xi^2yi
1	0,05	0,956	0,0025	0,000125	6,25E-06	0,0478	0,00239
2	0,15	0,832	0,0225	0,003375	0,000506	0,1248	0,01872
3	0,46	0,571	0,2116	0,097336	0,044775	0,26266	0,120824
4	0,7	0,378	0,49	0,343	0,2401	0,2646	0,18522
5	0,82	0,306	0,6724	0,551368	0,452122	0,25092	0,205754
6	1,17	0,104	1,3689	1,601613	1,873887	0,12168	0,142366
sum	3,35	3,147	2,7679	2,596817	2,611396	1,07246	0,675274
matriks:	6	3,35	2,7679	.	a1	=	3,147
	3,35	2,7679	2,596817		a2		1,07246
	2,7679	2,596817	2,611396		a3		0,675274
		A					B
A^{-1}	=	0,875541	-2,81916	1,87541	Pakai MINVERSE		
		-2,81916	14,46594	-11,3971			
		1,87541	-11,3971	9,728563			
a1	=	0,998304	Pakai MMULT				
a2		-1,0539					
a3		0,248467					
f(x) =	0,998304	+	-1,0539 x	+	0,248467 x^2		

3.1.4 Penjelasan

Program untuk metode regresi numerik di atas baik menggunakan python maupun excel sama-sama melakukan algoritma yang sama seperti pada dasar teori. Untuk mencari $a_1 + a_2x + a_3x^2$ menggunakan matriks A mencari terlebih dahulu Sum-sum yang diperlukan yaitu x, x^2, x^3, x^4 didapatkan nilainya 3.35, 2.7679, 2.597, 2.611. Kemudian mencari nilai matriks B menghitung sum-sum untuk y, xy, x^2y didapatkan 3.147, 1.072, 0.675. Kemudian invers matriks A dan kalikan dengan matriks B. Sehingga didapatkan nilai matriks a_1, a_2, a_3 untuk $a_1x^0 + a_2x^1 + a_3x^2$.

3.2 Penyelesaian dengan Metode Eliminasi Gauss

3.1.1 Source Code

```
# Metode Regresi
# 24060119130045 - Nashirudin Baqiy

import numpy as np
import pandas as pd

#data pada tabel
x = [0.05, 0.15, 0.46, 0.70, 0.82, 1.17]
y = [0.956, 0.832, 0.571, 0.378, 0.306, 0.104]
n = len(x)

def xn(x,n):
    s = x
    for i in range(n-1):
        s = s*x
    return s
def mat(x, y):
    #Panjang/jumlah data point
    n = len(x)
    #Inisiasi tabel
    A = [[None for x in range(4)] for x in range(3)]
    A[0][0] = n
    #Sum x
    sx = 0
    for i in range(n):
        sx = sx + x[i]
    A[1][0] = sx
    A[0][1] = sx
    #Sum x2
    sx2 = 0
    for i in range(n):
        sx2 = sx2 + xn(x[i],2)
    A[2][0] = sx2
    A[1][1] = sx2
    A[0][2] = sx2
    #Sum x3
    sx3 = 0
    for i in range(n):
        sx3 = sx3 + xn(x[i],3)
    A[2][1] = sx3
    A[1][2] = sx3
    #Sum x4
    sx4 = 0
    for i in range(n):
        sx4 = sx4 + xn(x[i],4)
    A[2][2] = sx4

    #Sum y
```

```
    sy = 0
    for i in range(n):
        sy = sy + y[i]
    A[0][3] = sy
    #Sum x.y
    sxy = 0
    for i in range(n):
        sxy = sxy + x[i]*y[i]
    A[1][3] = sxy
    #Sum x2y
    sx2y = 0
    for i in range(n):
        sx2y = sx2y + xn(x[i],2)*y[i]
    A[2][3] = sx2y

    return A

def gauss(A):
    n = len(A)
    tp = [[0 for i in range(n+1)] for i in range(n)]
    for m in range(0, n):
        for r in range(m, n):
            lead = A[r][m]
            for k in range(0, n+1):
                A[r][k] = A[r][k]/lead
        for r in range(n-1, -1, -1):
            for k in range(0, n+1):
                if r > m:
                    A[r][k] = A[r][k] - A[r-1][k]
        G_table = pd.DataFrame(A)
        print(G_table)
        print()

    x = [0 for i in range(n)]
    for i in range(n - 1, -1, -1):
        x[i] = A[i][n] / A[i][i]
        for k in range(i - 1, -1, -1):
            A[k][n] -= A[k][i] * x[i]

    return x

A = mat(x,y)
A_table = pd.DataFrame(A)
print(A_table)
print()

G = gauss(A)
print('persamaan polinomial: %f + %fx + %fx^2' % (G[0],G[1],G[2]))
```

3.1.2 Hasil Command Prompt

```
(base) F:\TUGAS\Praktikum\Metnum\Tugas3\py>gauss.py
        0         1         2         3
0  6.0000  3.350000  2.767900  3.147000
1  3.3500  2.767900  2.596817  1.072460
2  2.7679  2.596817  2.611396  0.675274

     0         1         2         3
0  1.0  0.558333  0.461317  0.524500
1  0.0  0.267905  0.313853 -0.204363
2  0.0  0.111952  0.168288 -0.076171

     0         1         2         3
0  1.0  0.558333  0.461317  0.524500
1  0.0  1.000000  1.171505 -0.762816
2  0.0  0.000000  0.331719  0.082421

     0         1         2         3
0  1.0  0.558333  0.461317  0.524500
1  0.0  1.000000  1.171505 -0.762816
2  0.0  0.000000  1.000000  0.248467

persamaan polinomial: 0.998304 + -1.053896x + 0.248467x^2
```

3.1.3 Excel

ELIMINASI GAUSS : f(x) = y = a + bx + cx^2

i	xi	yi	xi^2	xi^3	xi^4	xi.yi	xi^2yi
1	0,05	0,956	0,0025	0,000125	6,25E-06	0,0478	0,00239
2	0,15	0,832	0,0225	0,003375	0,000506	0,1248	0,01872
3	0,46	0,571	0,2116	0,097336	0,044775	0,26266	0,120824
4	0,7	0,378	0,49	0,343	0,2401	0,2646	0,18522
5	0,82	0,306	0,6724	0,551368	0,452122	0,25092	0,205754
6	1,17	0,104	1,3689	1,601613	1,873887	0,12168	0,142366
sum	3,35	3,147	2,7679	2,596817	2,611396	1,07246	0,675274

matriks:

$$\underbrace{\begin{vmatrix} 6 & 3,35 & 2,7679 \\ 3,35 & 2,7679 & 2,596817 \\ 2,7679 & 2,596817 & 2,611396 \end{vmatrix}}_{A} \cdot \begin{vmatrix} a \\ b \\ c \end{vmatrix} = \underbrace{\begin{vmatrix} 3,147 \\ 1,07246 \\ 0,675274 \end{vmatrix}}_{B}$$

M1:	1	0,558333	0,461317	0,5245
	0	0,267905	0,313853	-0,20436
	0	0,111952	0,168288	-0,07617
M2:	1	0,558333	0,461317	0,5245
	0	1	1,171505	-0,76282
	0	0	0,331719	0,082421
M3:	1	0,558333	0,461317	0,5245
	0	1	1,171505	-0,76282
	0	0	1	0,248467

f(x) = 0,998304 + -1,0539 x + 0,248467 x^2

3.1.4 Penjelasan

Program untuk metode eliminasi gauss di atas baik menggunakan python maupun excel sama-sama melakukan algoritma yang sama seperti pada dasar teori. Untuk mencari $a_1 + a_2x + a_3x^2$ menggunakan matriks A mencari terlebih dahulu Sum-sum yang diperlukan yaitu $x, x^2, x^3, x^4,\ y, xy, x^2y$ didapatkan 3.35, 2.7679, 2.597, 2.611, 3.147, 1.072, 0.675. Kemudian perulangan (M) agar lead 1 diagonal yaitu membagi baris yang akan menjadi lead 1 dengan bilangan itu sendiri lalu dikurangi dari bawah antara baris bawah dengan baris atasnya tetapi ada syarat untuk pengurangan yaitu row>M. Terjadi perulangan sebanyak n = 3, sehingga dapat mencari a_1, a_2, a_3 untuk $a_1x^0 + a_2x^1 + a_3x^2$

Nashirudin Baqiy
24060119130045
Lab A1

BAB IV
PENUTUPAN

4.1 Kesimpulan

Solusi model regresi polinomial dicari dengan Metode Regresi dan Metode Eliminasi Gauss. Langkah-langkah yang tersedia dapat dibuat dengan menggunakan Bahasa *Python* dan menggunakan Spreadsheet seperti excel sehingga tidak perlu melakukan iterasi secara sistem manual seperti yang ada pada *proses* di atas. Hasil kerja dari sistem program tersebut adalah sebagai berikut.

1. Pada soal tentang implementasi Metode Regresi, dicari solusi model regresi polinomial. Hasil yang diperoleh menggunakan Metode Regresi yang telah dibuat dalam Bahasa *Python* dan Excel adalah $y = 0.998 - 1.539x + 0.249x^2$.
2. Pada soal tentang implementasi Metode Eliminasi Gauss, dicari solusi model regresi polinomial. Hasil yang diperoleh menggunakan Metode Regresi yang telah dibuat dalam Bahasa *Python* dan Excel adalah $y = 0.998 - 1.539x + 0.249x^2$.