# Evaluasi Penyelenggara Diklat Sivia Studio 2022

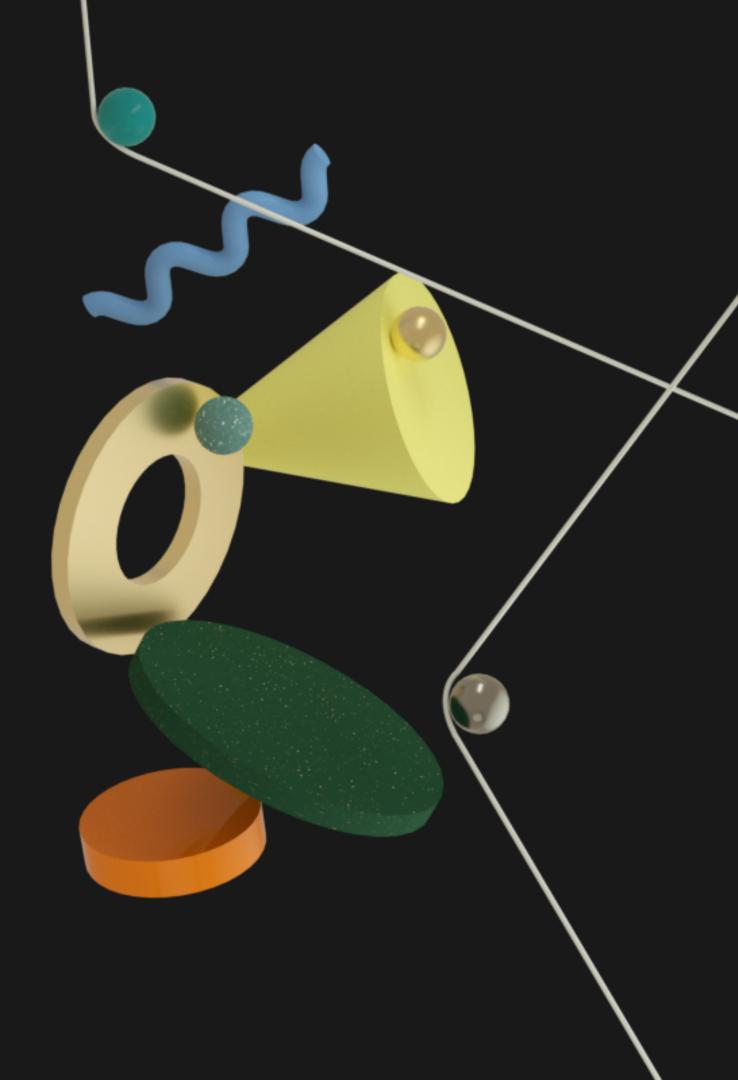
Non-Linear Programming (Riset Operasi)

Muhammad Aris Margono Yudha Islami Sulistya



# Dataset

Data Set	Multivariate	Number of	67
Characteristics:		Instances:	
Attribute	Integer.Real	Number of	10
Characteristics:		Attributes:	9 (X)
			1 (y)
Associated	Regression	Missing	No
Tasks:		Values?	



# Informasi Atribut



- 1. Efektifitas penyelenggara
- 2. Kesiapan dan kesediaan sarana diklat
- 3. Kesesuaian Pelaksanaan Program Dengan Rencana
- 4. Kebersihan kelas, asrama, kafetaria, toilet dll
- 5. Ketersediaan dan kelengkapan bahan diklat
- 6. Ketersediaan fasilitas olahraga, kesehatan dan ibadah
- 7. Pelayanan terhadap peserta dan widyaiswara
- 8. Administrasi diklat
- 9. Jenis Diklat
- 10. Evaluasi keseluruhan penyelenggara diklat



### Problem

Mencari Nilai Optimal Untuk Konstanta A,B,C,D,E,F,G,H,I dan Menghitung Evaluasi Berdasarkan Tingkat Akurasi Data Prediksi dan Data Aktual

#### **Tentukan Model Non Linear**

 $Y = A*X1^B*X2^c*X3^D*X4^E*X5^f*X6^G*X7^H*X8^T$ 

#### **Evaluasi Error**

- 1.MSE
- 2. RMSE
- 3. MAE
- 4. SE
- 5. R2 SCORE



# 1. Import Library

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # berfungsi untuk mengolah data numerik
import matplotlib.pyplot as plt # visualisasi data
import seaborn as sns # visualisasi data

from IPython.display import display # memunculkan data dalam bentuk tabel
from scipy.optimize import minimize # untuk mengoptimalkan fungsi
```

### 2. Membaca Dataset

```
df = pd.read_excel('evaluasi_penyelenggara_diklat_2022.xlsx') # berfungsi untuk membaca file excel
display(df.head()) # berfungsi untuk menampilkan 5 data teratas
display(df.tail()) # berfungsi untuk menampilkan 5 data terbawah
```

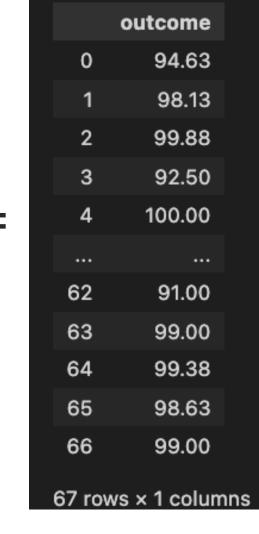
	1	2	3	4	5	6	7	8	9	outcome
0	95	98	98	98	98	85	95	90	1	94.63
1	95	95	100	95	100	100	100	100	1	98.13
2	100	100	100	99	100	100	100	100	1	99.88
3	95	80	95	90	95	95	95	95	1	92.50
4	100	100	100	100	100	100	100	100	1	100.00
	1	2	3	4	5	6	7	8	9	outcome
62	1 80	<b>2</b> 89	<b>3</b> 90	<b>4</b> 100	<b>5</b> 100	<b>6</b> 80	<b>7</b> 89	<b>8</b> 100	<b>9</b> 3	<b>outcome</b> 91.00
62 63										
	80	89	90	100	100	80	89	100	3	91.00
63	80 99	89 99	90 99	100 99	100 99	80 99	89 99	100 99	3	91.00 99.00

### 3. Feature Selection

```
# Memisahkan data menjadi 2 bagian yaitu X dan y atau data feature dan data target (outcome)
X = df.drop([9, 'outcome'], axis=1) # berfungsi untuk memisahkan data feature
y = df['outcome'] # berfungsi untuk memisahkan data target

display(X) # berfungsi untuk menampilkan data feature
display(pd.DataFrame(y)) # berfungsi untuk menampilkan data target
```

	0	95	98	98	98	98	85	95	90
	1	95	95	100	95	100	100	100	100
	2	100	100	100	99	100	100	100	100
	3	95	80	95	90	95	95	95	95
X =	4	100	100	100	100	100	100	100	100
<b>/</b> -									
	62	80	89	90	100	100	80	89	100
	63	99	99	99	99	99	99	99	99
	64	100	95	100	100	100	100	100	100
	65	99	99	95	99	99	99	99	100
	66	99	99	99	99	99	99	99	99
	67 rows × 8 columns								



## 4. Modelling

#### 4.1 Menentukan Variabel X1 - X8

```
xm1 = np.array(X[1]) # berfungsi untuk mengubah data feature menjadi array untuk Efektifitas penyelenggara
xm2 = np.array(X[2]) # berfungsi untuk mengubah data feature menjadi array untuk Kesiapan dan kesediaan sarana diklat
xm3 = np.array(X[3]) # berfungsi untuk mengubah data feature menjadi array untuk Kesesuaian Pelaksanaan Program Dengan Rencana
xm4 = np.array(X[4]) # berfungsi untuk mengubah data feature menjadi array untuk Kebersihan kelas, asrama, kafetaria, toilet dll
xm5 = np.array(X[5]) # berfungsi untuk mengubah data feature menjadi array untuk Ketersediaan dan kelengkapan bahan diklat
xm6 = np.array(X[6]) # berfungsi untuk mengubah data feature menjadi array untuk Ketersediaan fasilitas olahraga, kesehatan dan ibadah
xm7 = np.array(X[7]) # berfungsi untuk mengubah data feature menjadi array untuk Pelayanan terhadap peserta dan widyaiswara
xm8 = np.array(X[8]) # berfungsi untuk mengubah data feature menjadi array untuk Administrasi diklat
ym = y # berfungsi untuk mengubah data target menjadi array sebagai evaluasi keseluruhan penyelenggara diklat
```

### 4.2 Membuat Fungsi Model

```
def calc_y(x): # berfungsi untuk menghitung nilai y
    a = x[0] # berfungsi untuk mengambil nilai x pada indeks ke 0
    b = x[1] # berfungsi untuk mengambil nilai x pada indeks ke 1
    c = x[2] # berfungsi untuk mengambil nilai x pada indeks ke 2
    d = x[3] # berfungsi untuk mengambil nilai x pada indeks ke 3
    e = x[4] # berfungsi untuk mengambil nilai x pada indeks ke 4
    f = x[5] # berfungsi untuk mengambil nilai x pada indeks ke 5
    g = x[6] # berfungsi untuk mengambil nilai x pada indeks ke 6
    h = x[7] # berfungsi untuk mengambil nilai x pada indeks ke 7
    i = x[8] # berfungsi untuk mengambil nilai x pada indeks ke 8
    y = a * (xm1 ** b) * (xm2 ** c) * (xm3 ** d) * (xm4 ** e) * (xm5 ** f) * (xm6 ** g) * (xm7 ** h) * (xm8 ** i) # NON LINEAR - MODEL SATU

# y = a + ((b*xm1)) + ((c*xm2) ** 2) + ((d*xm3) ** 3) + ((e*xm4) ** 4) + ((f*xm5) ** 5) + ((g*xm6) ** 6) + ((h*xm7) ** 7) + ((i*xm8) ** 8) # NON LINEAR - MODEL DUA

return y # berfungsi untuk mengembalikan nilai y
```

#### 4.3 Menentukan Variabel X1 - X8

```
my_bnds = (-100, 100) # berfungsi untuk menginisialisasi nilai my_bnds
bnds = (my_bnds, my_bnds, my_bnds, my_bnds, my_bnds, my_bnds, my_bnds, my_bnds) # berfungsi untuk menginisialisasi nilai bnds
solution = minimize(objective, x0, method='SLSQP', bounds=bnds) # berfungsi untuk menghitung nilai solusi dengan metode SLSQP dan batasan bnds
x = solution.x # berfungsi untuk menghitung nilai x
y = calc_y(x) # berfungsi untuk menghitung nilai y prediksi
```

#### 4.4 Solusi Konstanta A-I

#### A = 1.0774766587568623

B = 0.13832903434061383

C = 0.1154091721956655

D = 0.11634977744074691

E = 0.1272001673682369

F = 0.1347280849503186

G = 0.11729324839522494

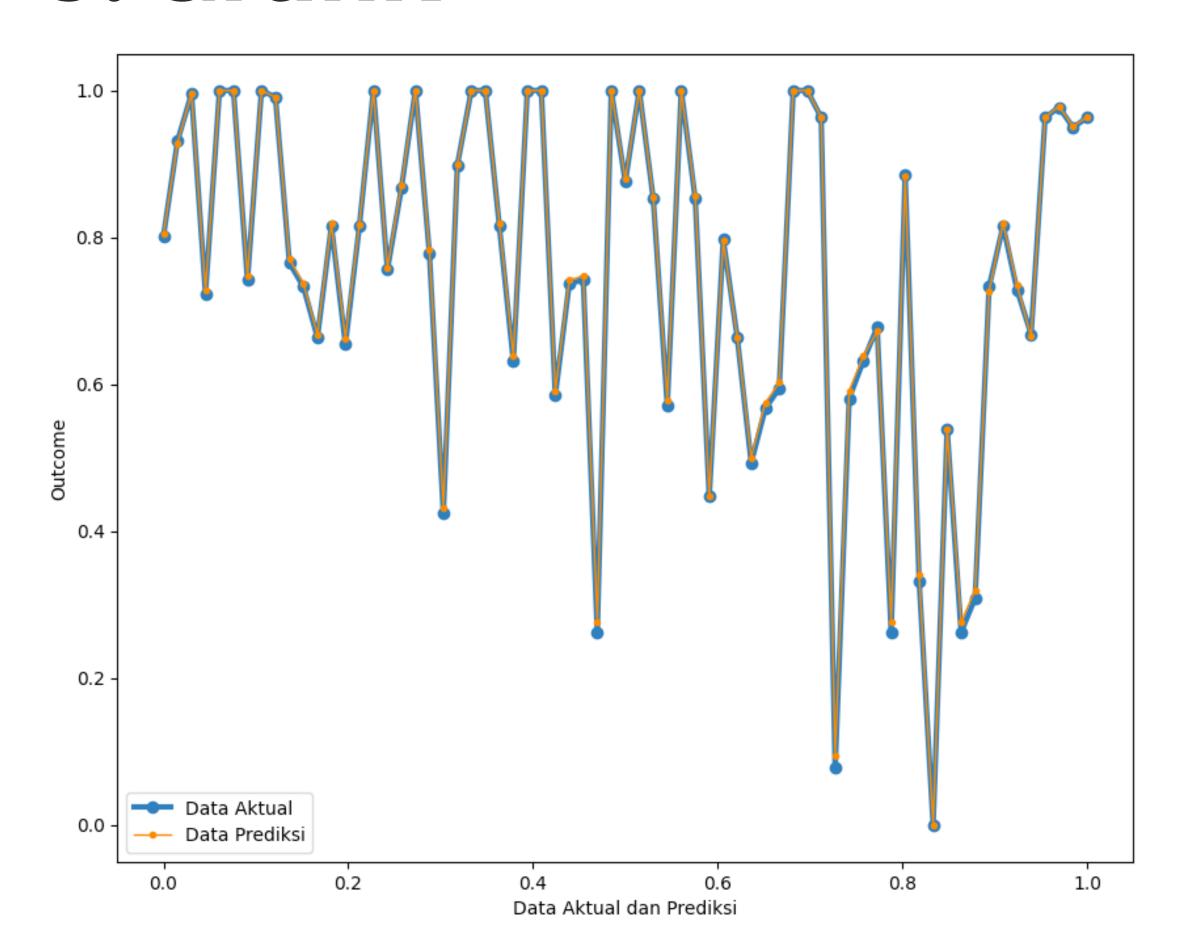
H = 0.11087449841576665

I = 0.12325272799118284

### 4.5 Hasil y\_true dan y\_pred

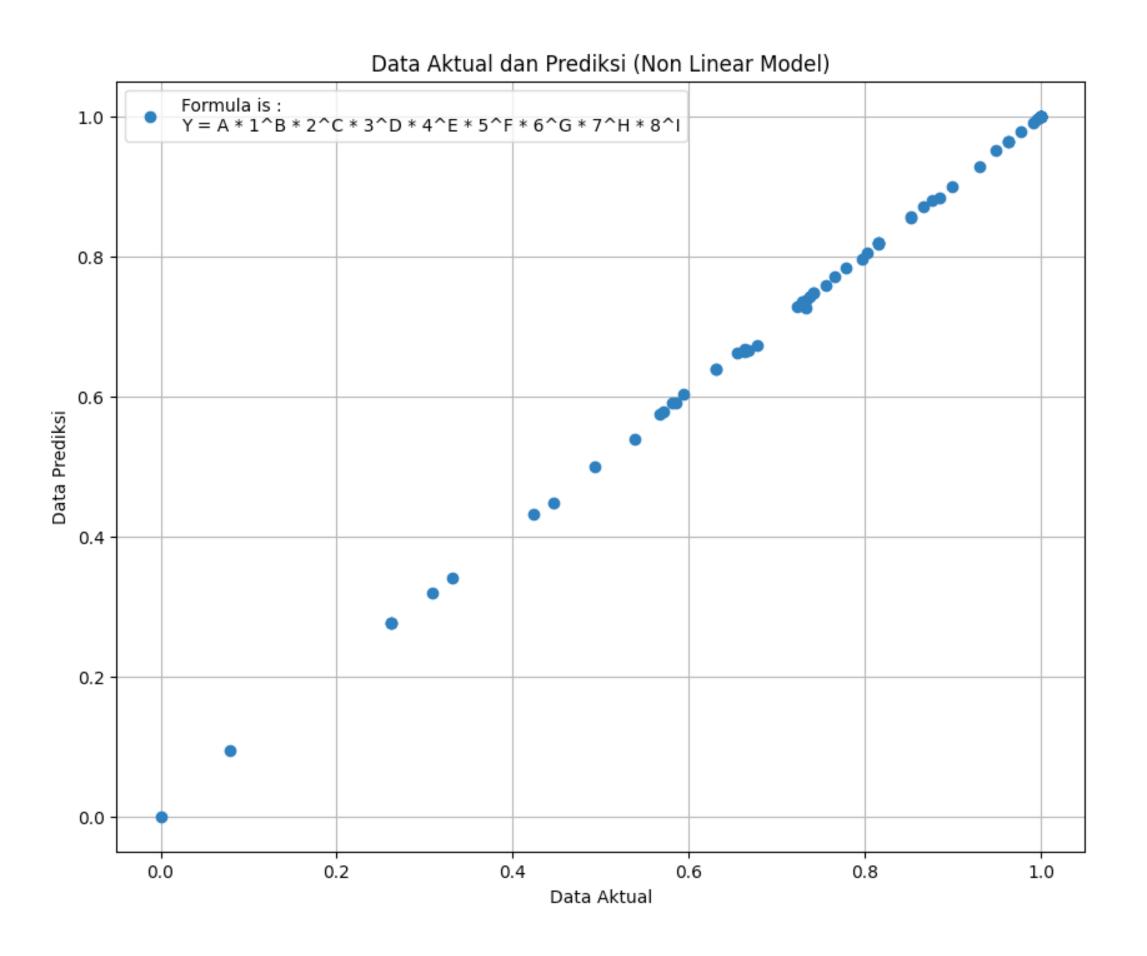
	Data Aktual	Data Prediksi
0	94.63	94.510475
1	98.13	97.902856
2	99.88	99.707101
3	92.50	92.421639
4	100.00	99.834648
62	91.00	90.726226
63	99.00	98.852756
64	99.38	99.245401
65	98.63	98.501478
66	99.00	98.852756
67 rov	ws × 2 columns	<b>;</b>

### 5. Grafik





5.1 Grafik Data Aktual dan Data Prediksi





5.2 Grafik Data Aktual dan Data Prediksi (Non Linear Model)

### 6. Evaluasi

### 6.1. Membuat Fungsi Error & Error<sup>2</sup>

```
def error(y_true, y_pred): # berfungsi untuk menghitung nilai error
    return y_true - y_pred # berfungsi untuk menghitung nilai error dengan rumus y_true - y_pred

def error2(y_true, y_pred): # berfungsi untuk menghitung nilai error2
    return (y_true - y_pred)**2 # berfungsi untuk menghitung nilai error2 dengan rumus (y_true - y_pred)^2
```

```
final['error'] = error(final['y_true'], final['y_pred']) # berfungsi untuk menghitung nilai error
final['error'] = error2(final['y_true'], final['y_pred']) # berfungsi untuk menghitung nilai error2
total_error = final['error'].sum() # berfungsi untuk menghitung nilai total error
display(final) # berfungsi untuk menampilkan nilai final
```

### 6.1. Hasil Evaluasi

	y_true	y_pred	error	error²				
0	94.63	94.510475	0.119525	0.014286				
1	98.13	97.902856	0.227144	0.051595				
2	99.88	99.707101	0.172899	0.029894				
3	92.50	92.421639	0.078361	0.006140				
4	100.00	99.834648	0.165352	0.027341				
62	91.00	90.726226	0.273774	0.074952				
63	99.00	98.852756	0.147244	0.021681				
64	99.38	99.245401	0.134599	0.018117				
65	98.63	98.501478	0.128522	0.016518				
66	99.00	98.852756	0.147244	0.021681				
67 ro	67 rows × 4 columns							



#### 6.1. Hasil Evaluasi

```
MSE = total_error / len(final) # berfungsi untuk menghitung nilai MSE dengan rumus total error dibagi dengan panjang final

RMSE = np.sqrt(MSE) # berfungsi untuk menghitung nilai RMSE dengan rumus akar kuadrat dari MSE

MAE = final['error'].abs().sum() / len(final) # berfungsi untuk menghitung nilai MAE dengan rumus total error absolut dibagi dengan panjang final

SE = final['error'].std() # berfungsi untuk menghitung nilai SE dengan rumus standar deviasi dari error

R2SCORE = 1 - (final['error'].sum() / ((final['y_true'] - final['y_true'].mean())**2).sum()) # berfungsi untuk menghitung nilai R2SCORE dengan rumus 1 - (total error kuadrat dibagi dengan ((y_true - rata-rata y_true)^2).sum())

print('MSE = ' + str(MSE)) # berfungsi untuk menampilkan nilai MSE

print('RMSE = ' + str(RMSE)) # berfungsi untuk menampilkan nilai RMSE

print('MAE = ' + str(MSE)) # berfungsi untuk menampilkan nilai MAE

print('SE = ' + str(SE)) # berfungsi untuk menampilkan nilai SE

print('R2 SCORE = ' + str(R2SCORE)) # berfungsi untuk menampilkan nilai R2SCORE
```

```
MSE = 0.020985240344601547

RMSE = 0.1448628328613021

MAE = 0.12391725296085078

SE = 0.11269384025207152

R2 SCORE = 0.9995202005308449
```

## Kesimpulan

#### Soluasi Variabel A-I

A = 1.0774766587568623

B = 0.13832903434061383

C = 0.1154091721956655

D = 0.11634977744074691

E = 0.1272001673682369

F = 0.1347280849503186

G = 0.11729324839522494

H = 0.11087449841576665

I = 0.12325272799118284

#### **Evaluasi Error**

MSE = 0.020985240344601547

RMSE = 0.1448628328613021

MAE = 0.12391725296085078

SE = 0.11269384025207152

R2 SCORE = 0.9995202005308449

## THANKS

Muhammad Aris Margono Yudha Islami Sulistya