



Data Normalization

Nadila Imaarah

3323600015

2 D4 Sains Data Terapan A

Lets Roll

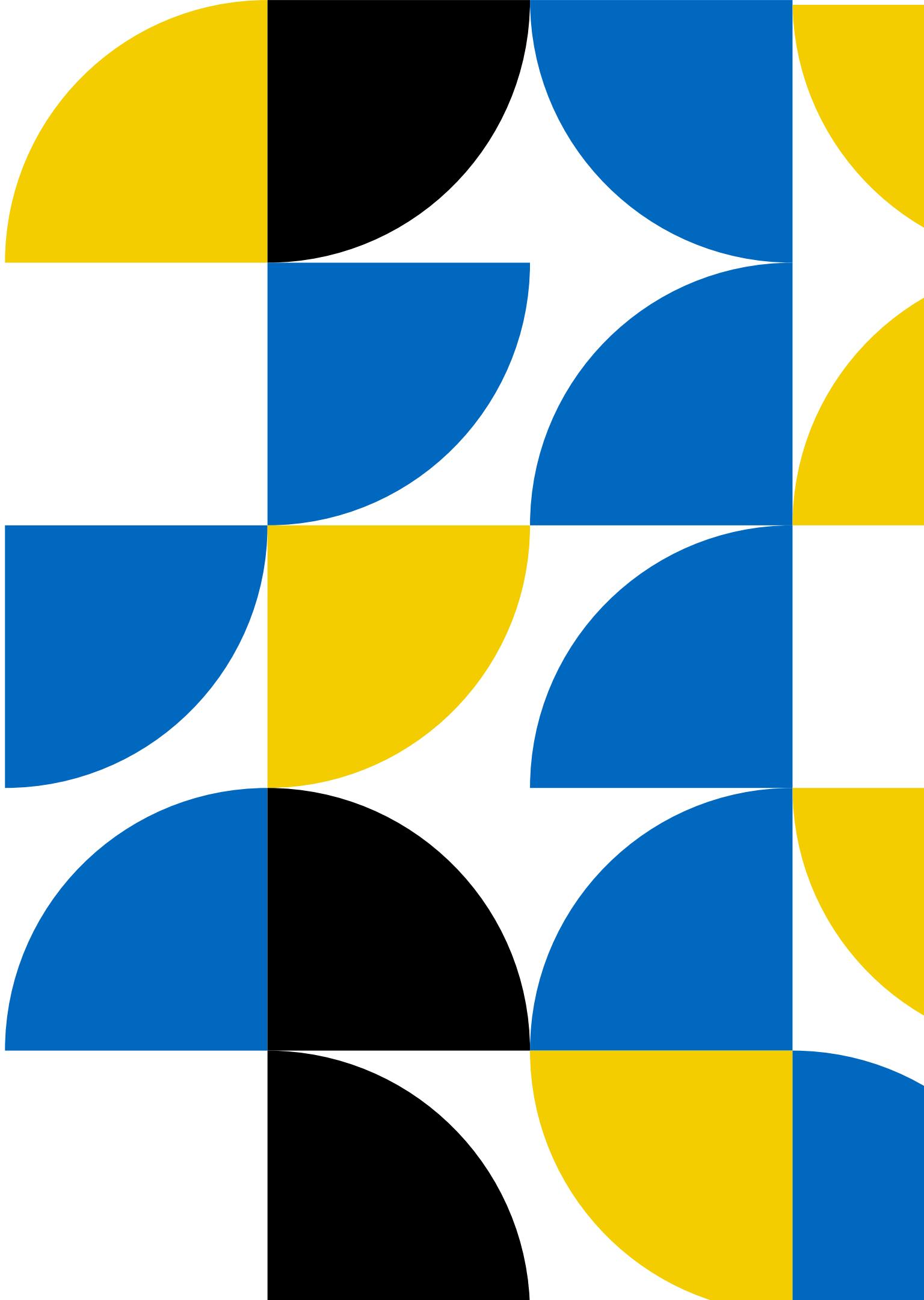


Table Of Content

- 
- 01 Dataset Ruspini
 - 02 Menampilkan Dataset Ruspini
 - 03 Pengisian Nilai Missing Values
 - 04 Mencari Nilai Statistik pada Atribut
 - 05 Dataset Titanic
 - 06 Menampilkan jumlah baris dan kolom
 - 07 Mengambil fitur Age dan Fare
 - 08 Mengambil fitur survived

Table Of Content

- 
- 09 Mengisi missing values
 - 10 Normalisasi min-max dengan library
 - 11 Normalisasi min-max secara manual
 - 12 Normalisasi z-score dengan library
 - 13 Normalisasi z-score secara manual
 - 14 Normalisasi sigmoidal dengan library
 - 15 Normalisasi sigmoidal secara manual
 - 16 Link Koding di Colab



Dataset Ruspini

Pada studi kasus pertama ini akan menggunakan dataset Ruspini.

Data ruspini ini adalah dataset yang terdiri dari fitur x,fitur y dan fitur label.
Dataset Ruspini ini sering digunakan untuk dilakukan clustering



Dataset Ruspini

	x	y	label
0	4	53	1
1	5	63	1
2	10	59	1
3	9	77	1
4	13	49	1
...
70	66	23	4
71	61	25	4
72	76	27	4
73	72	31	4
74	64	30	4
75 rows × 3 columns			

```
● ● ●  
1 import pandas as pd  
2 import numpy as np  
3 df = pd.read_csv('ruspini.csv')  
4 df
```

Pada tahap pertama ini
saya load dataset ruspini
dan menampilkan dataset
ruspini yang akan saya
lakukan analisis lebih lanjut



Pengisian Nilai Missing Values

Dataset			
	x	y	label
0	4.0	53	1
1	NaN	63	1
2	10.0	59	1
3	9.0	77	1
4	13.0	49	1
..
70	66.0	23	4
71	61.0	25	4
72	NaN	27	4
73	72.0	31	4
74	64.0	30	4

```
● ● ●
1 import pandas as pd
2
3 # Membaca dataset dari file CSV
4 dataset = pd.read_csv('ruspini_missing.csv')
5 # Menampilkan dataset sebelum pengisian missing value
6 print('Dataset\n', dataset)
7 # Mengisi nilai yang hilang dengan rata-rata berdasarkan grup 'CLASS'
8 dataset = dataset.fillna(dataset.groupby('label').transform('mean'))
9 # Menampilkan dataset setelah pengisian missing value
10 print('\n\nDataset setelah pengisian missing value\n', dataset)
11
```

```
Dataset setelah pengisian missing value
      x    y  label
0  4.000000  53     1
1  20.764706  63     1
2  10.000000  59     1
3  9.000000  77     1
4  13.000000  49     1
..    ...
70 66.000000  23     4
71 61.000000  25     4
72 72.000000  27     4
73 72.000000  31     4
74 64.000000  30     4
[75 rows x 3 columns]
```

Pada tahap ini saya menampilkan dataset yang mengandung nilai NaN. Setelah itu dilakukan handling missing values, di sini saya menggunakan metode mean untuk mengisi nilai NaN. Setelah dilakukan handling missing values disini saya menampilkan dataset disebelah kanan,tampak tidak ada data yang mengandung NaN

Mencari Nilai Statistikal



```
1 import pandas as pd
2 import numpy as np
3 df = pd.read_csv('ruspini.csv')
4 df_x= df['x']
5
6 minval = df_x.min()
7 maxval = df_x.max()
8 avgval = df_x.mean()
9 stdval = df_x.std()
10
11 print('nilai minimum =', minval)
12 print('nilai maksimum =', maxval)
13 print('nilai rata-rata =', avgval)
14 print('nilai deviasi standar =', stdval)
```

```
nilai minimum = 4
nilai maksimum = 117
nilai rata-rata = 54.88
nilai deviasi standar = 30.50252980203
```

Mencari nilai statistik dari atribut x, yang terdiri dari nilai minimum, nilai maksimum, nilai rata-rata atau mean, dan standar deviasi untuk mengetahui distribusi data pada fitur x



Dataset Titanic

Dataset Titanic ini saya dapatkan dari web Kaggle. Data ini berisi informasi mengenai penumpang pada kapal Titanic.

Pada dataset ini terdiri dari beberapa variable yakni pclass,survived,name,sex,fare dan lain sebagainya



Speed

Speed in data collection increases



Processing

Data processing becomes more effective

Menampilkan Dataset

```
● ● ●  
1 import pandas as pd  
2 import matplotlib.pyplot as plt  
3 import seaborn as sns  
4 df = pd.read_csv('titanic.csv')  
5 df.head()
```

Selanjutnya disini saya menggunakan Data Titanic. Disini saya meload dan menampilkan dataset. Pada dataset titanic ini terdapat beberapa fitur yakni pclass,survived,name,sex,fare dan lain sebagainya

Ouput

pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON

Menampilkan Jumlah Baris Dan kolom

```
● ● ●  
1 rows, cols = df.shape  
2 print(f"Jumlah baris: {rows}, Jumlah kolom: {cols}")
```

OUTPUT

```
Jumlah baris: 1349, Jumlah kolom: 14
```



Pada tahap ini saya menampilkan jumlah baris dan jumlah kolom pada dataset titanic. Berdasarkan output didapatkan bahwa jumlah baris sebanyak 1349 dan jumlah kolomnya sebanyak 14

Mengambil dan Menampilkan Fitur Age dan Fare



```
1 data = pd.DataFrame(df,columns=['age' , 'fare'])  
2 data.head()
```

Pada tahap ini saya menampilkan kolom age dan fare untuk mengecek apakah fitur ini benar ada di dalam dataset titanic yang saya gunakan serta menampilkan 5 data teratas

	age	fare
0	29.0000	211.3375
1	0.9167	151.5500
2	2.0000	151.5500
3	30.0000	151.5500
4	25.0000	151.5500

Mengambil dan Menampilkan Fitur Survived



```
1 data = pd.DataFrame(df,columns= ['survived'])  
2 data.head()
```

	survived
0	1
1	1
2	0
3	0
4	0

Pada tahap ini saya menampilkan fitur survived. Fitur survived berisikan nilai 0 dan 1 dimana nilai 0 menandakan not survived dan nilai 1 menandakan survived

HANDLING MISSING VALUE PADA FITUR AGE



Menampilkan Missing Value Pada Kolom Age



```
1 print("Jumlah missing value fitur 'Age' :\n", df['age'].isna().sum())
```

```
Jumlah missing value fitur 'Age'
```

```
273
```

Pada tahap ini saya menampilkan jumlah missing value pada fitur age. Pada fitur age didapatkan ada 273 data yang missing value

Handling Missing Value

Pada fitur Age

```
● ● ●  
1 import pandas as pd  
2 # Mengisi nilai yang hilang dalam kolom 'Age' berdasarkan rata-rata grup 'Survived'  
3 df['age'] = df.groupby('survived')['age'].transform(lambda x: x.fillna(x.mean()))  
4 # Menampilkan hasil setelah pengisian missing values  
5 print("\nDataset setelah pengisian missing value:\n", df[['age', 'survived']])  
6
```

Pada tahap ini saya mengisi missing value pada fitur age yang bertipe data numerik, untuk handling missing values ini saya menggunakan metode mean dari masing-masing kelas survived.menggunakan groupby('survived').transform('mean')

Dataset setelah pengisian missing value

	age	survived
0	29.000000	1
1	0.916700	1
2	2.000000	0
3	30.000000	0
4	25.000000	0
...
1344	14.500000	0
1345	29.920694	0
1346	26.500000	0
1347	27.000000	0
1348	29.000000	0

Memastikan Sudah Tidak Ada Missing Value

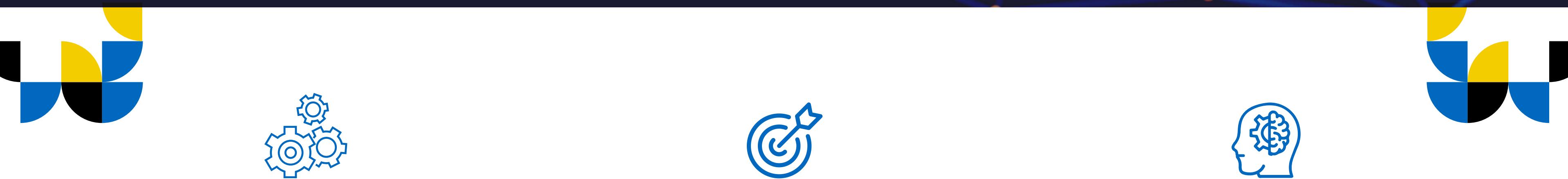


```
1 # Menampilkan jumlah missing value setelah pengisian  
2 print("\nJumlah missing value fitur age setelah handling missing value : \n", df['age'].isna().sum())
```

```
Jumlah missing value fitur age setelah handling missing value :  
0
```

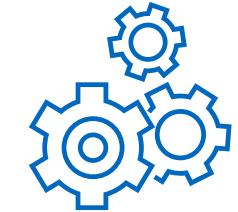
Setelah melakukan handling missing value di sini saya menampilkan kembali apakah masih ada missing value pada fitur fage dan didapatkan sudah tidak ada missing value

NORMALISASI



Min-Max Scaler

Mengubah nilai dengan rentang 0 hingga 1



Z-Score atau Standard Scaler

Merubah nilai agar data memiliki mean 0 dan standar deviasi 1



Sigmoidal Scaler

merubah nilai dengan rentang 0 hingga 1 dengan memperhatikan nilai eksponensialnya

Menghapus Kolom Yang Tidak Dipakai Untuk Normalisasi

```
● ● ●  
1 # Daftar kolom yang akan dihapus  
2 kolom_dihapus = ['pclass', 'survived', 'name', 'sex', 'sibsp', 'parch',  
3                 'ticket', 'cabin', 'embarked', 'boat', 'body', 'home.dest']  
4 # Menghapus kolom-kolom tersebut dari DataFrame  
5 df_norm = df.drop(columns=kolom_dihapus)  
6 # Menampilkan hasil  
7 print("Kolom setelah penghapusan:", df_norm.columns.tolist())  
8 print(df_norm.head())  
9
```

OUTPUT

```
Kolom setelah penghapusan: ['age', 'fare']  
      age      fare  
0  29.0000  211.3375  
1   0.9167  151.5500  
2   2.0000  151.5500  
3  30.0000  151.5500  
4  25.0000  151.5500
```



Pada tahap ini saya menghapus beberapa fitur dan hanya menyisahkan fitur age dan fare untuk normalisasi. Saya hanya memilih fitur age dan fare untuk normalisasi karena nilai nya memiliki rentang yang cukup jauh

Normalisasi Min-Max dengan Library



```
1 minmax_scaler = MinMaxScaler()  
2 df_minmax = pd.DataFrame(minmax_scaler.fit_transform(df_norm), columns=df_norm.columns)  
3 print("Normalisasi Min-Max:\n", df_minmax)
```

Pada Tahap ini saya melakukan normalisasi dengan menggunakan min-maz scaller dengan bantuan library

```
Normalisasi Min-Max:  
      age      fare  
0   0.361169  0.412503  
1   0.009395  0.295806  
2   0.022964  0.295806  
3   0.373695  0.295806  
4   0.311064  0.295806  
...   ...     ...  
1344  0.179540  0.028213  
1345  0.372702  0.028213  
1346  0.329854  0.014102  
1347  0.336117  0.014102  
1348  0.361169  0.015371  
  
[1349 rows x 2 columns]
```

Normalisasi Min-Max Secara Manual



```
1 df = pd.DataFrame(df_norm)
2 # Normalisasi manual menggunakan Min-Max Scaling
3 def min_max_normalize(column):
4     return (column - column.min()) / (column.max() - column.min())
5 # Menerapkan normalisasi ke setiap kolom
6 df_normalized_manual = df.apply(min_max_normalize)
7 # Menampilkan hasil normalisasi
8 print("Dataset setelah normalisasi:\n", df_normalized_manual)
```

Pada Tahap ini saya melakukan normalisasi dengan menggunakan min-max scaller secara manual, untuk memudahkanya saya membuat suatu fungsi bernama `min_max_normalize`.

Dataset setelah normalisasi:

	age	fare
0	0.361169	0.412503
1	0.009395	0.295806
2	0.022964	0.295806
3	0.373695	0.295806
4	0.311064	0.295806
...
1344	0.179540	0.028213
1345	0.378875	0.028213
1346	0.329854	0.014102
1347	0.336117	0.014102
1348	0.361169	0.015371

[1349 rows x 2 columns]

Output yang dihasilkan normalisasi secara manual memiliki hasil yang sama dengan output dengan normalisasi dengan bantuan library

Normalisasi Z-Score dengan Library

```
● ● ●  
1 zscore_scaler = StandardScaler()  
2 df_zscore = pd.DataFrame(zscore_scaler.fit_transform(df_norm), columns=df_norm.columns)  
3 print("\nNormalisasi Z-Score:\n", df_zscore)
```

Pada tahap ini saya melakukan normalisasi dengan z-score,dimana normalisasi ini merubah nilai agar data memiliki mean 0 dan standar deviasi 1

Normalisasi Z-Score:

	age	fare
0	-7.204044e-02	3.495574
1	-2.269441e+00	2.325659
2	-2.184677e+00	2.325659
3	6.205365e-03	2.325659
4	-3.850237e-01	2.325659
...
1344	-1.206605e+00	-0.357017
1345	5.559699e-16	-0.357017
1346	-2.676550e-01	-0.498477
1347	-2.285321e-01	-0.498477
1348	-7.204044e-02	-0.485758

Normalisasi Z-Score secara manual

```
● ● ●  
1 df = pd.DataFrame(df_norm)  
2 # Standardisasi Z-Score manual  
3 def z_score_normalize(column):  
4     return (column - column.mean()) / column.std()  
5 # Menerapkan standardisasi Z-Score ke setiap kolom  
6 df_zscore_manual = df.apply(z_score_normalize)  
7 # Menampilkan hasil standardisasi  
8 print("Dataset setelah standardisasi Z-Score:\n", df_zscore_manual)
```

```
Dataset setelah standardisasi Z-Score:  
          age      fare  
0    -0.074568  3.494278  
1    -2.270718  2.324797  
2    -2.186003  2.324797  
3     0.003633  2.324797  
4    -0.387373  2.324797  
...       ...     ...  
1344 -1.208487 -0.356885  
1345  0.035975 -0.356885  
1346 -0.270071 -0.498292  
1347 -0.230971 -0.498292  
1348 -0.074568 -0.485578  
  
[1349 rows x 2 columns]
```

Pada Tahap ini saya melakukan normalisasi dengan menggunakan z-score scaller secara manual, untuk memudahkanya saya membuat suatu fungsi bernama `z_score_normalize`.

Output yang dihasilkan normalisasi secara manual memiliki hasil yang sedikit berbeda dengan output dengan normalisasi dengan bantuan library, hal ini disebabkan karena perhitungan standar deviasi, pada normalisasi dengan library `StandardScaler()` perhitungan standar deviasi menggunakan populasi, sedangkan normalisasi manual menggunakan library `pandas`, `pandas` ini menggunakan standar deviasi sampel, selain itu terdapat perbedaan dalam menangani data. Pada `StandardScaler()` mengubah data ke dalam bentuk `NumPy array` sebelum melakukan perhitungan, sedangkan `Pandas` melakukan operasi secara kolom per kolom.

Normalisasi Sigmoidal dengan Library

```
● ● ●  
1 from scipy.special import expit  
2 import pandas as pd  
3 df = pd.DataFrame(df_norm)  
4 # Menerapkan fungsi sigmoid ke setiap elemen di DataFrame  
5 df_sigmoid = df.applymap(expit)  
6 print(df_sigmoid)
```

Pada tahap ini saya melakukan normalisasi dengan menggunakan library . Didapatkan output bahwa nilai pada fitur age dan fare berada di rentang 0 hingga 1

	age	fare
0	1.000000	1.000000
1	0.714369	1.000000
2	0.880797	1.000000
3	1.000000	1.000000
4	1.000000	1.000000
...
1344	0.999999	0.999999
1345	1.000000	0.999999
1346	1.000000	0.999272
1347	1.000000	0.999272
1348	1.000000	0.999620

Normalisasi Sigmoidal Secara Manual

```
● ● ●  
1 df = pd.DataFrame(df_norm)  
2 # Normalisasi Sigmoid manual  
3 def sigmoid_normalize(column):  
4     return 1 / (1 + np.exp(-column))  
5 # Menerapkan normalisasi Sigmoid ke setiap kolom  
6 df_sigmoid_manual = df.apply(sigmoid_normalize)  
7 # Menampilkan hasil normalisasi  
8 print("Dataset setelah normalisasi Sigmoid:\n", df_sigmoid_manual)
```

```
Dataset setelah normalisasi Sigmoid:  
      age      fare  
0    1.000000  1.000000  
1    0.714369  1.000000  
2    0.880797  1.000000  
3    1.000000  1.000000  
4    1.000000  1.000000  
...   ...   ...  
1344 0.999999  0.999999  
1345 1.000000  0.999999  
1346 1.000000  0.999272  
1347 1.000000  0.999272  
1348 1.000000  0.999620  
  
[1349 rows x 2 columns]
```

Pada Tahap ini saya melakukan normalisasi dengan menggunakan sigmoidal secara manual, untuk memudahkanya saya membuat suatu fungsi bernama sigmoid_normalize.

Output yang dihasilkan normalisasi secara manual memiliki hasil yang sama dengan output dengan nromalisasi dengan bantuan library



Link Colab

Program Python Tugas Normalisasi



Speed

Speed in data collection increases



Processing

Data processing becomes more effective

...

Thank You