

Catatan :

- Sebelum memulai coding pastikan sudah terinstall dipython library Numpy dan Sklearn.
- Untuk Langkah kerja dan hasil percobaan, SS an nya di pisah per Teknik Preprocessing.
- Analisis tidak boleh ada yang sama, apabila kedapatan sama maka untuk laporan praktikum selanjutnya akan dikerjakan dengan metode tulis tangan atau tidak di ketik.
- Pada Langkah kerja kodingannya di cantumkan nama dan stambuk masing-masing atau di buat penanda agar berbeda dengan yang lain.

Modul IV
Kecerdasan Buatan
Teknik Preprocessing Data dengan Python

I. Tujuan

1. Mahasiswa mampu memahami tahapan-tahapan yang ada pada preprocessing data.
2. Mampu membuat program sederhana teknik preprocessing data.

II. Alat Dan Bahan

1. Masukkan sesuai alat dan bahan yang kalian miliki
2. Dst...

III. Teori Dasar

Dalam kehidupan kita sehari-hari, kita berurusan dengan banyak data tetapi data ini dalam bentuk mentah. Untuk menyediakan data sebagai input dari algoritma machine learning, kita perlu mengubahnya menjadi data yang bermakna. Di situlah preprocessing data muncul. Dengan kata sederhana lainnya, kita dapat mengatakan bahwa sebelum memberikan data ke algoritma machine learning, kita perlu memproses data terlebih dahulu. Berikut adalah beberapa teknik-teknik dalam preprocessing data:

1. Binarization atau Binarisasi Binarisasi adalah teknik preprocessing yang digunakan ketika kita perlu mengubah nilai numerik kita menjadi nilai Boolean. Kita dapat menggunakan metode inbuilt untuk binerisasi data input katakanlah dengan menggunakan 0,5 sebagai nilai ambang batas.
2. Mean Removal Ini adalah teknik preprocessing lain yang sangat umum yang digunakan dalam machine learning. Pada dasarnya digunakan untuk menghilangkan mean dari vektor fitur sehingga setiap fitur berpusat pada nol. Kami juga dapat menghilangkan bias dari fitur dalam vektor fitur.
3. Scaling atau Penskalaan Ini adalah teknik preprocessing data lain yang digunakan untuk menskalakan vektor fitur. Penskalaan vektor fitur diperlukan karena nilai setiap fitur dapat bervariasi di antara banyak nilai acak. Dengan kata lain, kita dapat mengatakan bahwa penskalaan itu

penting karena kita tidak ingin fitur apa pun menjadi besar atau kecil secara sintetis.

4. Normalization Normalisasi adalah teknik preprocessing data lain yang digunakan untuk memodifikasi vektor fitur. Modifikasi semacam itu diperlukan untuk mengukur vektor fitur pada skala umum.

IV. Langkah Kerja

```
import numpy as np
from sklearn import preprocessing

#input data
input_data = np.array([[2.1, -1.9, 5.5], [-1.5, 2.4, 3.5], [0.5, -7.9, 5.6], [5.9, 2.3, -5.8]])
print(input_data)

#binarisasi
data_binarized = preprocessing.Binarizer(threshold=0.5).transform(input_data)
print("\nBinarized Data:\n", data_binarized)

#mean removal
print("Mean =", input_data.mean(axis=0))
print("Std deviation =", input_data.std(axis=0))

data_scaled = preprocessing.scale(input_data)
print("Mean =", data_scaled.mean(axis=0))
print("Std deviation =", data_scaled.std(axis=0))

#min max scaling
data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(0,1))
data_scaled_minmax = data_scaler_minmax.fit_transform(input_data)
print("\nMin max scaled data:\n", data_scaled_minmax)

#L1 normalisasi
data_normalized_l1 = preprocessing.normalize(input_data, norm='l1')
print("\nL1 Normalized data:\n", data_normalized_l1)
#L2 normalisasi
data_normalized_l2 = preprocessing.normalize(input_data, norm='l2')
print("\nL2 Normalized data:\n", data_normalized_l2)
```

V. Hasil Percobaan

```

Binarized Data:
[[1. 0. 1.]
 [0. 1. 1.]
 [0. 0. 1.]
 [1. 1. 0.]]
Mean = [ 1.75 -1.275  2.2  ]
Std deviation = [2.71431391 4.20022321 4.69414529]
Mean = [1.11022302e-16 0.00000000e+00 0.00000000e+00]
Std deviation = [1. 1. 1.]

Min max scaled data:
[[0.48648649 0.58252427 0.99122807]
 [0.          1.          0.81578947]
 [0.27027027 0.          1.          ]
 [1.          0.99029126 0.          ]]

```

```

L1 Normalized data:
[[ 0.22105263 -0.2          0.57894737]
 [-0.2027027  0.32432432  0.47297297]
 [ 0.03571429 -0.56428571  0.4          ]
 [ 0.42142857  0.16428571 -0.41428571]]

L2 Normalized data:
[[ 0.33946114 -0.30713151  0.88906489]
 [-0.33325106  0.53320169  0.7775858  ]
 [ 0.05156558 -0.81473612  0.57753446]
 [ 0.68706914  0.26784051 -0.6754239  ]]

```

VI. Analisis

VII. Kesimpulan