

Machine Learning

Pertemuan 2



Dibuat oleh :
Dennis Parulian Panjaitan
2241720182
Teknik Informatika-3A

**PRODI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI
MALANG 2024**

1. Implementasi Normalisasi

```
1  import numpy as np
2  from sklearn.preprocessing import MinMaxScaler
3
4  np.set_printoptions(precision=6) # bulatkan 4 angka koma
5  np.set_printoptions(suppress=True) # bulatkan 4 angka koma
6
7  # Kita akan membentuk data
8  # Hal ini dikarenakan, scikit-learn hanya menerima input
9  # dalam bentuk n-dimensional array
10 data = [
11     [100, 0.0001],
12     [50, 0.05],
13     [30, 0.003]
14 ]
15
16 # Ubah ke bentuk numpy n-dimensional array
17 data = np.asarray(data)
18 print('Data Asli')
19 print(data)
20
21 # Mendefinisikan obyek MinMaxScaler
22 scaler = MinMaxScaler ()
23 # Transformasikan data
24 scaled = scaler.fit_transform(data)
25 print ('Data Normalisasi' )
26 print (scaled)
```

Hasil :

```
TTTT/Machine Learning/Pert
Data Asli
[[100.      0.0001]
 [ 50.      0.05 ]
 [ 30.      0.003 ]]
Data Normalisasi
[[1.      0.      ]
 [0.285714 1.      ]
 [0.      0.058116]]
```

2. Implementasi Standarisasi

```
1 import numpy as np
2 from sklearn.discriminant_analysis import StandardScaler
3
4 np.set_printoptions(precision=6) # bulatkan 4 angka koma
5 np.set_printoptions(suppress=True) # bulatkan 4 angka koma
6
7 # Kita akan membentuk data
8 # Hal ini dikarenakan, scikit-Learn hanya menerima input
9 # dalam bentuk n-dimensional array
10 data = [
11     [100, 0.0001],
12     [50, 0.05],
13     [30, 0.003]
14 ]
15
16 # Ubah ke bentuk numpy n-dimensional array
17 data = np.asarray(data)
18 print('Data Asli')
19 print(data)
20
21 # Mendefinisikan obyek MinMaxScaler
22 scaler = StandardScaler ()
23 # Transformasikan data
24 scaled = scaler.fit_transform(data)
25 print ('Data Standarisasi' )
26 print (scaled)
```

Hasil :

```
[-1.019049 -0.642757]]
PS C:\kuliah\SEMESTER 5
Data Asli
[[100.      0.0001]
 [ 50.      0.05 ]
 [ 30.      0.003 ]]
Data Standarisasi
[[ 1.358732 -0.76956 ]
 [-0.339683  1.412317]
 [-1.019049 -0.642757]]
PS C:\kuliah\SEMESTER 5
```

3. Implementasi Ordinal Encoding

```
akukun5.py / ...  
  
from sklearn.preprocessing import OrdinalEncoder  
  
# Inisiasi obyek Ordinal Encoder  
oe = OrdinalEncoder ()  
  
# Definisikan dataa  
# dalam bentuk 2d  
data = [  
    ['Politeknik Negeri Malang' ],  
    ['Politeknik Elektronika Negeri Surabaya'],  
    ['Politeknik Negeri Jakarta'],  
    ['Politeknik Negeri Semarang' ]  
]  
  
# Transformasi Ordinal Encoder  
transform_oe = oe.fit_transform(data)  
  
print('Data Asli')  
print(data)  
  
print('Data Transformasi Ordinal Encoder')  
print(transform_oe)
```

Hasil :

```
[[['Politeknik Negeri Malang'], ['Politeknik Elektronika Negeri Surabaya'], ['Politeknik Negeri Jakarta'], ['Politeknik Negeri Semarang']]]  
Data Transformasi Ordinal Encoder  
[[2.]  
 [0.]  
 [1.]  
 [3.]]
```

4. Implementasi One-Hot Encoding

```
from sklearn.preprocessing import OneHotEncoder

# Inisiasi obyek Ordinal Encoder
ohe = OneHotEncoder ()

# Definisikan dataa
# dalam bentuk 2d
data = [
    ['Politeknik Negeri Malang' ],
    ['Politeknik Elektronika Negeri Surabaya'],
    ['Politeknik Negeri Jakarata'],
    ['Politeknik Negeri Semarang' ]
]

# Transformasi Ordinal Encoder
transform_ohe = ohe.fit_transform(data)

print('Data Asli')
print(data)

print('Data Transformasi One-Hot Encoding')
print(transform_ohe.toarray())
```

Hasil :

```
Data Asli
[['Politeknik Negeri Malang'], ['Politeknik Elektronika Negeri Surabaya'], ['Politeknik Negeri Jakarata'], ['Politeknik Negeri Semarang']]
Data Transformasi One-Hot Encoding
[[0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]]
```

Dr. G. Muliaji (CONCEPT & IMPLEMENTATION Machine Learning) Berkebangsaan 2021

5. Implementasi Dummy Variable Encoding

```
from sklearn.preprocessing import OneHotEncoder

# Inisiasi obyek Ordinal Encoder
de = OneHotEncoder (drop='first')

# Definisikan dataa
# dalam bentuk 2d
data = [
    ['Politeknik Negeri Malang' ],
    ['Politeknik Elektronika Negeri Surabaya'],
    ['Politeknik Negeri Jakarta'],
    ['Politeknik Negeri Semarang' ]
]

# Transformasi Ordinal Encoder
transform_de = de.fit_transform(data)

print('Data Asli')
print(data)

print('Data Transformasi One-Hot Encoding')
print(transform_de.toarray())
```

Hasil :

```
Data Asli
[['Politeknik Negeri Malang'], ['Politeknik Elektronika Negeri Surabaya'], ['Politeknik Negeri Jakarta'], ['Politeknik Negeri Semarang']]
Data Transformasi One-Hot Encoding
[[0. 1. 0.]
 [0. 0. 0.]
 [1. 0. 0.]
 [0. 0. 1.]]
```

D:\C:\Vuliah\SEMESTER 5\NTATTTTTT\Machine Learning\Portofolio 2\

6. Studi Kasus Ekstraksi Fitur dari Data Teks

```
corpus = [
    'the house had a tiny little mouse',
    'the cat saw the mouse',
    'the mouse ran away from the house',
    'the cat finally ate the mouse',
    'the end of the mouse story'
]

from sklearn.feature_extraction.text import TfidfVectorizer

# Inisiasi obyek TfidfVectorizer
vect = TfidfVectorizer(stop_words='english' )

# Pembobotan TF-IDF
resp = vect.fit_transform(corpus)

# Cetak hasil
print('Hasil TF-IDF')
print(resp)

# Cetak token hasil stopwords
print('Hasil Token  ')
print(vect.get_feature_names_out())
```

Hasil :

```
Hasil TF-IDF
(0, 7)      0.2808823162882302
(0, 6)      0.5894630806320427
(0, 11)     0.5894630806320427
(0, 5)      0.47557510189256375
(1, 9)      0.7297183669435993
(1, 2)      0.5887321837696324
(1, 7)      0.3477147117091919
(2, 1)      0.5894630806320427
(2, 8)      0.5894630806320427
(2, 7)      0.2808823162882302
(2, 5)      0.47557510189256375
(3, 0)      0.5894630806320427
(3, 4)      0.5894630806320427
(3, 2)      0.47557510189256375
(3, 7)      0.2808823162882302
(4, 10)     0.6700917930430479
(4, 3)      0.6700917930430479
(4, 7)      0.3193023297639811
Hasil Token
['ate' 'away' 'cat' 'end' 'finally' 'house' 'little' 'mouse' 'ran' 'saw'
 'story' 'tiny']
```

Tugas praktikum :

1. Salin kalimat pada Kode 1-7 dengan tanda baca titik pada setiap kalimatnya dengan menggunakan editor teks.
2. Simpan kalimat tersebut pada file '.txt' dengan nama 'corpus.txt'.
3. Lakukan proses ekstraksi fitur TF-IDF dengan menggunakan file 'corpus.txt'.

Jawab :

```
gas.py / ...  
from sklearn.feature_extraction.text import TfidfVectorizer  
  
# Baca file corpus.txt  
with open('corpus.txt', 'r') as file:  
    corpus = file.readlines()  
  
# Hapus newline dari setiap baris  
corpus = [line.strip() for line in corpus]  
  
# Inisiasi obyek TfidfVectorizer  
vect = TfidfVectorizer(stop_words='english')  
  
# Pembobotan TF-IDF  
resp = vect.fit_transform(corpus)  
  
# Cetak Hasil  
print(resp)  
  
print(vect.get_feature_names_out())
```

```
corpus.txt  
1 the house had a tiny little mouse.  
2 the cat saw the mouse.  
3 the mouse ran away from the house.  
4 the cat finally ate the mouse.  
5 the end of the mouse story.  
6
```

Hasil :


```
13 0. (Kilian) SEMESTER 3 NDA111111 Machine Learning (CET exam) 27 & 0.70313702
(0, 7) 0.2808823162882302
(0, 6) 0.5894630806320427
(0, 11) 0.5894630806320427
(0, 5) 0.47557510189256375
(1, 9) 0.7297183669435993
(1, 2) 0.5887321837696324
(1, 7) 0.3477147117091919
(2, 1) 0.5894630806320427
(2, 8) 0.5894630806320427
(2, 7) 0.2808823162882302
(2, 5) 0.47557510189256375
(3, 0) 0.5894630806320427
(3, 4) 0.5894630806320427
(3, 2) 0.47557510189256375
(3, 7) 0.2808823162882302
(4, 10) 0.6700917930430479
(4, 3) 0.6700917930430479
(4, 7) 0.3193023297639811
['ate' 'away' 'cat' 'end' 'finally' 'house' 'little' 'mouse' 'ran' 'saw'
'story' 'tiny']
```