Pre-Processing-Feature Scaling

April 4, 2020

LALU ACHMAD WIRAHARLAN - 5170411207

[18]:
```
from pandas import DataFrame
import pandas as pd
import numpy as np
import math
from sklearn.preprocessing import LabelEncoder
```

[11]:
```
df = pd.read_csv("Apartemen.csv",header=0)
print(df)
```

```
   KodeApt  Wilayah  St_Milik  Jum_Kamar
0    104.0        0         1        3.0
1    197.0        2         0        3.0
2   8837.0        2         0        3.0
3    201.0        3         0        1.0
4    203.0        3         1        3.0
5    207.0        3         1        3.0
6    837.0        1         1        2.0
7    213.0        0         1        3.0
8    215.0        0         1        3.0
```

[12]:
```
print(df.describe())
```

```
           KodeApt   Wilayah  St_Milik  Jum_Kamar
count     9.000000  9.000000  9.000000   9.000000
mean   1223.777778  1.555556  0.666667   2.666667
std    2863.130445  1.333333  0.500000   0.707107
min     104.000000  0.000000  0.000000   1.000000
25%     201.000000  0.000000  0.000000   3.000000
50%     207.000000  2.000000  1.000000   3.000000
75%     215.000000  3.000000  1.000000   3.000000
max    8837.000000  3.000000  1.000000   3.000000
```

[13]:
```
df.loc[2,'Jum_Kamar'] = 100
print(df)
print(df.shape)
print(df.Jum_Kamar)
```

```
   KodeApt  Wilayah  St_Milik  Jum_Kamar
0    104.0        0         1        3.0
1    197.0        2         0        3.0
2   8837.0        2         0      100.0
3    201.0        3         0        1.0
4    203.0        3         1        3.0
5    207.0        3         1        3.0
6    837.0        1         1        2.0
7    213.0        0         1        3.0
8    215.0        0         1        3.0
(9, 4)
0      3.0
1      3.0
2    100.0
3      1.0
4      3.0
5      3.0
6      2.0
7      3.0
8      3.0
Name: Jum_Kamar, dtype: float64
```

[14]:
```
data = np.array(df)   #konversi data csv menjadi array
data = data.astype(float)   #konversi data menjadi tipe float
n_data = len(data[:,0]) #menghitung banyaknya data

n_feature = len(data[0,:])
print(n_feature)
```

```
4
```

[15]:
```
#min-max normalization
for i in range(0,n_feature):
    data[:,i] = ((data[:,i] - min(data[:,i]))/(max(data[:,i])-min(data[:,i])))

print(data)
```

```
[[0.         0.         1.         0.02020202]
 [0.01064926 0.66666667 0.         0.02020202]
 [1.         0.66666667 0.         1.        ]
 [0.01110729 1.         0.         0.        ]
 [0.01133631 1.         1.         0.02020202]
 [0.01179434 1.         1.         0.02020202]
 [0.0839345  0.33333333 1.         0.01010101]
 [0.01248139 0.         1.         0.02020202]
 [0.01271041 0.         1.         0.02020202]]
```

Pada pre-processing diatas ini menggunakan metode normalisasi min-max dengan rentang nilai

[0,1]. Data setiap atribut dilakukan pemrosesan dengan perulangan for sehingga seluruh data ternormalisasi. Rumus yang digunakan dengan mengubah index menjadi array lalu dilakukan pemrosesan berikut data = ((data - min(data[:,i]))/(max(data[:,i]) - min(data[:,i]))

[10]:
```
#Z-score normalization
data = np.array(df)   #konversi data csv menjadi array
data = data.astype(float)   #konversi data menjadi tipe float
n_data = len(data[:,0]) #menghitung banyaknya data

n_feature = len(data[0,:])
# print(data)

datamean = df.mean(axis=0)
# print(datamean)

datastd = df.std(axis=0)
# print(datastd)

df = (df - datamean)/(datastd)

print(df)
```

```
    KodeApt   Wilayah  St_Milik  Jum_Kamar
0 -0.391103 -1.166667  0.666667  -0.321706
1 -0.358621  0.333333 -1.333333  -0.321706
2  2.659055  0.333333 -1.333333   2.666054
3 -0.357224  1.083333 -1.333333  -0.383309
4 -0.356525  1.083333  0.666667  -0.321706
5 -0.355128  1.083333  0.666667  -0.321706
6 -0.135089 -0.416667  0.666667  -0.352508
7 -0.353032 -1.166667  0.666667  -0.321706
8 -0.352334 -1.166667  0.666667  -0.321706
```

Pada pre-processing diatas ini menggunakan metode normalisasi Z-Score. Data setiap atribut dilakukan pemrosesan normalisasi sehingga seluruh data ternormalisasi. Pada data ini tidak diubah menjadi array, namun tetap berbentuk dataFrame seperti awal, setelah itu dilakukan normalisasi dengan formula df = (df - datamean)/(datastd). df merupakan data awal yang belum ternormalisasi, datamean merupakan variable penyimpan data yang telah melalui pemrosesan mean. datastd merupakan varibale penyimpan data hasil pemrosesan standar deviasi.