preprocessing

March 19, 2022

1 Tugas 3 Data Mining - Preprocessing

Nama : Wahyu Adi Nugroho NIM : A11.2019.12310 Kelp : A11.46UG

Kerjakan Latihan tahapan preprocessing data pada slide sebelumnya, dapat menggunakan dataset yang lain / dimodifikasi, simpan dalam preprocessing.py atau preprocessing.ipynb, repositorikan file pada github.com dan kirimkan URL github melalui Assignment pada kulino (Pada blok Minggu ke-3).

1.1 1. Import Library Utama

[1]:

```
import numpy as np # mengolah matrix
import matplotlib.pyplot as plt # membuat visualisasi data : diagram
import pandas as pd # mengambil, mengolah, memanipulasi data
```

1.2 2. Import Dataset

[2]:

```
data_parpol = pd.read_csv('data-parpol.csv')
x = data_parpol.iloc[:, :-1].values # attr1, attr2, attr3
y = data_parpol.iloc[:, -1].values  # label
```

[3]:

```
print(x)
print(y)
```

```
[['PDIP' 15.0 27000000.0]
 ['PKS' 4.0 11000000.0]
 ['PSI' 0.0 2500000.0]
 ['Gerindra' 7.0 nan]
 ['Perindo' 0.0 3000000.0]
 ['PKPI' nan 1000000.0]
 ['PBB' 0.0 750000.0]
 ['Golkar' 8.0 20000000.0]]
['Ya' 'Ya' 'Tidak' 'Ya' 'Tidak' 'Tidak' 'Tidak' 'Ya']
```

1.3 3. Menghilangkan Missing Value

[4]:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
x[:, 1:3] = imputer.fit_transform(x[:, 1:3])
```

[5]:

```
print(x)
```

```
[['PDIP' 15.0 27000000.0]
 ['PKS' 4.0 11000000.0]
 ['PSI' 0.0 2500000.0]
 ['Gerindra' 7.0 9321428.57142857]
 ['Perindo' 0.0 3000000.0]
 ['PKPI' 4.857142857142857 1000000.0]
 ['PBB' 0.0 750000.0]
 ['Golkar' 8.0 20000000.0]]
```

1.4 4. Encoding data kategori (Atribut)

[6]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers = [('encoder', OneHotEncoder(), [0])],␣
 ↪remainder="passthrough")
x = np.array(ct.fit_transform(x))
```

[7]:

```
print(x)
```

```
[[0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 15.0 27000000.0]
 [0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 4.0 11000000.0]
 [0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 2500000.0]
 [1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 7.0 9321428.57142857]
 [0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 3000000.0]
 [0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 4.857142857142857 1000000.0]
 [0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 750000.0]
 [0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0 20000000.0]]
```

1.5 5. Encoding data kategori (Class / Label)

[8]:

```
print(y) # sebelum
```

```
['Ya' 'Ya' 'Tidak' 'Ya' 'Tidak' 'Tidak' 'Tidak' 'Ya']
```

[9]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

[10]:

```
print(y) #sesudah
```

```
[1 1 0 1 0 0 0 1]
```

1.6 6. Melakukan Feature Scaling

[11]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x[:, 8:] = sc.fit_transform(x[:, 8:])
```

[12]:

```
print(x)
```

```
[[0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 2.0875582115943256 1.9484301020685981]
 [0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -0.17641336999388676
  0.18500245413580632]
 [0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 -0.9996757632986912 -0.7518184838284893]
 [1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4410334249847166 0.0]
 [0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -0.9996757632986912 -0.6967113698305896]
 [0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 -1.8280097287100104e-16
  -0.9171398258221884]
 [0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 -0.9996757632986912 -0.9446933828211383]
 [0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.6468490233109176 1.1769305060980018]]
```

1.7 7. Membagi Dataset ke dalam Training dan Test Set

[13]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25,␣
 ↪random_state=1)
```

[14]:

```
print(x_train)
```

```
[[0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 -0.17641336999388676
  0.18500245413580632]
 [0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 -0.9996757632986912 -0.9446933828211383]
 [0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 2.0875582115943256 1.9484301020685981]
 [0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 -0.9996757632986912 -0.6967113698305896]
 [1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.4410334249847166 0.0]
 [0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 -1.8280097287100104e-16
  -0.9171398258221884]]
```

[15]:

```
print(x_test)
```

```
[[0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.6468490233109176 1.1769305060980018]
 [0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 -0.9996757632986912 -0.7518184838284893]]
```

[16]:

```
print(y_train)
```

```
[1 0 1 0 1 0]
```

[17]:

```
print(y_test)
```

```
[1 0]
```