tugas-praktikum

October 26, 2024

1 Tugas Praktikum (Wisconsin Breast Cancer)

Kelompok 1: 1. Arya Chandra 2. Elis Nurhidayati 3. Jantra Lang Lang B 4. Putri Ayu A

Deskripsi Tugas

Pada tugas pratikum ini Anda akan menggunakan data "Wisconsin Breast Cancer". Data tersebut terdiri dari 569 data yang digunakan untuk mendiagnonis jenis kanker Malignant (M) dan Benign (B). Tugas Anda adalah,

1. Pisahkan antara variabel yang dapat digunakan dan variabel yang tidak dapat digunakan.
2. Lakukan proses encoding pada kolom "diagnosis".
3. Lakukan proses standarisasi pada semua kolom yang memiliki nilai numerik.
4. Lakukan proses stratified split data untuk membuat data latih dan data uji dengan rasio 80:20.

##0 Load Data

[1]:
```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/Elis - 3C/SMT 5/Pembelajaran Mesin -␣
↪Ely Setyo Astuti, ST., M.T/Jobsheet/JS 2/wbc.csv')
df.head()
```

[1]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
0	0.11840	0.27760	0.3001	0.14710	
1	0.08474	0.07864	0.0869	0.07017	
2	0.10960	0.15990	0.1974	0.12790	
3	0.14250	0.28390	0.2414	0.10520	
4	0.10030	0.13280	0.1980	0.10430	

	…	texture_worst	perimeter_worst	area_worst	smoothness_worst	\
0	…	17.33	184.60	2019.0	0.1622	

```
1  …          23.41           158.80      1956.0            0.1238
2  …          25.53           152.50      1709.0            0.1444
3  …          26.50            98.87       567.7            0.2098
4  …          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

##1. Pisahkan antara variabel yang dapat digunakan dan variabel yang tidak dapat digunakan.

Drop kolom yang tidak digunakan yaitu id dan unnamed: 32

Alasan: Kolom id dan Unnamed: 32 tidak berguna untuk analisis karena hanya sebagai pengenal dan tidak memberikan informasi apa pun tentang diagnosis. Kolom Unnamed: 32 merupakan artefak dari proses pengumpulan data dan tidak berisi data yang berguna.

[2]:
```
df = df.drop(columns=['id', 'Unnamed: 32'])
df.head()
```

[2]:
```
  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0         M        17.99         10.38          122.80     1001.0
1         M        20.57         17.77          132.90     1326.0
2         M        19.69         21.25          130.00     1203.0
3         M        11.42         20.38           77.58      386.1
4         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0         0.2419  …         25.38          17.33           184.60
```

```
1         0.1812  …         24.99          23.41           158.80
2         0.2069  …         23.57          25.53           152.50
3         0.2597  …         14.91          26.50            98.87
4         0.1809  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 31 columns]
```

1.1 2. Lakukan proses encoding pada kolom "diagnosis"

Kolom diagnosis adalah variabel kategorikal dengan nilai 'M' (Malignant) dan 'B' (Benign). Model pembelajaran mesin biasanya membutuhkan input numerik, jadi kami menggunakan LabelEncoder untuk mengubah kategori ini menjadi angka (0 dan 1).

[4]:
```
from sklearn.preprocessing import LabelEncoder, StandardScaler

le = LabelEncoder() # membuat objek dari LabelEncoder
df['diagnosis'] = le.fit_transform(df['diagnosis']) # proses encoding

df.tail()
```

[4]:
```
     diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
564          1        21.56         22.39          142.00     1479.0
565          1        20.13         28.25          131.20     1261.0
566          1        16.60         28.08          108.30      858.1
567          1        20.60         29.33          140.10     1265.0
568          0         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000
```

```
     symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
564         0.1726  …        25.450          26.40           166.10
565         0.1752  …        23.690          38.25           155.00
566         0.1590  …        18.980          34.12           126.70
567         0.2397  …        25.740          39.42           184.60
568         0.1587  …         9.456          30.37            59.16

     area_worst  smoothness_worst  compactness_worst  concavity_worst  \
564      2027.0           0.14100            0.21130           0.4107
565      1731.0           0.11660            0.19220           0.3215
566      1124.0           0.11390            0.30940           0.3403
567      1821.0           0.16500            0.86810           0.9387
568       268.6           0.08996            0.06444           0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[5 rows x 31 columns]
```

Hasil proces encoding kolom diagnosis menampilkan nilai (1) untuk kategori 'M' (Malignant) dan (0) untuk kategoori 'B' (Benign)

##3. Lakukan proses standarisasi pada semua kolom yang memiliki nilai numerik.

Menstandarisasi semua kolom pada dataframe df, kecuali kolom 'diagnosis', menggunakan StandardScaler dari scikit-learn. Kolom yang dipilih akan ditransformasikan agar memiliki mean 0 dan standar deviasi 1.

[7]:
```
std = StandardScaler()
standartColumn = df.columns.difference(['diagnosis'])

df[standartColumn] = std.fit_transform(df[standartColumn])

df.head()
```

[7]:
```
   diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0          1     1.097064     -2.073335        1.269934   0.984375
1          1     1.829821     -0.353632        1.685955   1.908708
2          1     1.579888      0.456187        1.566503   1.558884
3          1    -0.768909      0.253732       -0.592687  -0.764464
4          1     1.750297     -1.151816        1.776573   1.826229

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
```

```
0         1.568466          3.283515        2.652874             2.532475
1        -0.826962         -0.487072       -0.023846             0.548144
2         0.942210          1.052926        1.363478             2.037231
3         3.283553          3.402909        1.915897             1.451707
4         0.280372          0.539340        1.371011             1.428493

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0       2.217515  …      1.886690      -1.359293         2.303601
1       0.001392  …      1.805927      -0.369203         1.535126
2       0.939685  …      1.511870      -0.023974         1.347475
3       2.867383  …     -0.281464       0.133984        -0.249939
4      -0.009560  …      1.298575      -1.466770         1.338539

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0    2.001237          1.307686           2.616665         2.109526
1    1.890489         -0.375612          -0.430444        -0.146749
2    1.456285          0.527407           1.082932         0.854974
3   -0.550021          3.394275           3.893397         1.989588
4    1.220724          0.220556          -0.313395         0.613179

   concave points_worst  symmetry_worst  fractal_dimension_worst
0              2.296076        2.750622                 1.937015
1              1.087084       -0.243890                 0.281190
2              1.955000        1.152255                 0.201391
3              2.175786        6.046041                 4.935010
4              0.729259       -0.868353                -0.397100

[5 rows x 31 columns]
```

##4. Lakukan proses stratified split data untuk membuat data latih dan data uji dengan rasio 80:20.

[ ]:
```python
# Split data
from sklearn.model_selection import train_test_split

# Split data training dan dan lainnya

df_train, df_test = train_test_split(df, test_size=0.2, random_state=0,␣
 ↪stratify=df['diagnosis'])

# Cek masing-masing ukuran data

print(f'Jumlah label data asli:\n{df.diagnosis.value_counts()}')
print(f'Jumlah label data train:\n{df_train.diagnosis.value_counts()}')
print(f'Jumlah label data test:\n{df_test.diagnosis.value_counts()}')
```

```
Jumlah label data asli:
diagnosis
```

```
0    357
1    212
Name: count, dtype: int64
Jumlah label data train:
diagnosis
0    285
1    170
Name: count, dtype: int64
Jumlah label data test:
diagnosis
0    72
1    42
Name: count, dtype: int64
```