Tugas Pertemuan Minggu ke - 5 - Klasifikasi dengan Naive Bayes

April 2, 2022

0.1 Tugas Pertemuan Minggu ke-5 - Klasifikasi dengan Naive Bayes

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

Kerjakan Latihan tahapan klasifikasi dengan naïve bayes pada latihan sebelumnya, dataset bisa diganti / dimodifikasi, simpan dalam naive_bayes.py atau naive_bayes.ipynb, repositorikan file pada github.com dan kirimkan URL github melalui Assignment pada kulino (Pada blok Minggu ke-5).

Link Github : https://github.com/wahyu-adi-n/data-mining-46ug/tree/main/T5

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
dataset = pd.read_excel("../DATASET/datapemilukpu.xls")
```

[3]:
```
dataset.head(5)
```

[3]:
```
   NAMA PARTAI POLITIK        NAMA CALON LEGESLATIF JENIS KELAMIN  KECAMATAN  \
0  HATI NURANI RAKYAT              TOTO SUKISNO,BSc             L   LEBAKSIU
1  HATI NURANI RAKYAT               EDI PURYANTO,SH             L      SLAWI
2  HATI NURANI RAKYAT              ELI RETNOWATI,SH             P      SLAWI
3  HATI NURANI RAKYAT                      SAHYUDIN             L  DUKUHWARU
4  HATI NURANI RAKYAT   H.FAJAR SIGIT KUSUMAJAYA,SH             L      SLAWI

   NO.URUT PARPOL  SUARA SAH PARTAI  JUML.PEROLEHAN KURSI  DAERAH PEMILIHAN  \
0               1             18578                     1                 1
1               1             18578                     1                 1
2               1             18578                     1                 1
3               1             18578                     1                 1
4               1             18578                     1                 2

   NO.URUT CALEG  SUARA SAH CALEG TERPILIH ATAU TIDAK
0            1.0              594               TIDAK
1            2.0              943               TIDAK
2            3.0             1730               TIDAK
3            4.0             2508                  YA
4            1.0              923               TIDAK
```

[4]:

```
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

[5]:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy="constant",␣
 ↪fill_value=1)
x[:, 8:9] = imputer.fit_transform(x[:, 8:9])
```

[6]:

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

[7]:

```
x = x[:, 8:10]
```

[8]:

```
print(x)
```

```
[[1.0 594]
 [2.0 943]
 [3.0 1730]
 [4.0 2508]
 [1.0 923]
 [2.0 308]
 [3.0 54]
 [1.0 1682]
 [2.0 918]
 [3.0 87]
 [1.0 728]
 [2.0 346]
 [3.0 184]
 [1.0 381]
 [2.0 148]
 [3.0 128]
 [1.0 441]
 [2.0 88]
 [3.0 74]
 [1.0 351]
 [1.0 972]
 [2.0 307]
 [1.0 274]
 [2.0 186]
 [1.0 844]
 [2.0 333]
 [1.0 804]
 [2.0 275]
 [1.0 848]
 [2.0 461]
 [3.0 119]
 [1.0 130]
```

```
[1.0 219]
[1.0 154]
[1.0 179]
[1.0 237]
[1.0 621]
[1.0 1912]
[1.0 265]
[2.0 117]
[3.0 110]
[4.0 124]
[1.0 504]
[2.0 500]
[3.0 286]
[4.0 237]
[1.0 3149]
[2.0 494]
[3.0 423]
[1.0 678]
[2.0 258]
[3.0 186]
[1.0 891]
[2.0 255]
[3.0 1328]
[1.0 1293]
[2.0 902]
[3.0 583]
[1.0 446]
[2.0 195]
[3.0 152]
[4.0 157]
[5.0 231]
[1.0 95]
[2.0 121]
[3.0 18]
[1.0 148]
[1.0 106]
[1.0 111]
[1.0 93]
[1.0 1697]
[2.0 790]
[3.0 432]
[4.0 979]
[5.0 841]
[6.0 485]
[7.0 1076]
[8.0 235]
[9.0 338]
[1.0 720]
```

```
[2.0 505]
[3.0 886]
[4.0 639]
[5.0 367]
[6.0 104]
[7.0 159]
[8.0 69]
[1.0 696]
[2.0 414]
[3.0 1297]
[4.0 1701]
[5.0 151]
[6.0 42]
[7.0 45]
[1.0 2656]
[2.0 926]
[3.0 177]
[4.0 234]
[5.0 312]
[6.0 387]
[7.0 575]
[8.0 165]
[9.0 388]
[10.0 281]
[1.0 1445]
[2.0 630]
[3.0 460]
[4.0 296]
[5.0 393]
[6.0 203]
[7.0 258]
[8.0 206]
[9.0 198]
[10.0 263]
[1.0 2081]
[2.0 412]
[3.0 1379]
[4.0 5214]
[5.0 454]
[6.0 203]
[7.0 203]
[8.0 2239]
[9.0 126]
[1.0 967]
[2.0 2067]
[3.0 116]
[4.0 1176]
[1.0 796]
```

```
[2.0 174]
[3.0 736]
[4.0 2615]
[1.0 1788]
[2.0 1297]
[3.0 389]
[4.0 2560]
[1.0 1213]
[2.0 171]
[3.0 1461]
[4.0 458]
[5.0 729]
[6.0 25]
[7.0 31]
[1.0 7765]
[2.0 3921]
[3.0 180]
[4.0 196]
[5.0 573]
[1.0 2251]
[2.0 2319]
[3.0 808]
[1.0 62]
[1.0 37]
[1.0 108]
[2.0 30]
[1.0 36]
[1.0 59]
[1.0 4622]
[2.0 2940]
[3.0 1350]
[4.0 391]
[5.0 360]
[6.0 980]
[7.0 1042]
[8.0 50]
[1.0 2219]
[2.0 6334]
[3.0 1165]
[4.0 5351]
[5.0 424]
[1.0 5006]
[2.0 762]
[3.0 1621]
[4.0 2951]
[5.0 2288]
[6.0 130]
[1.0 4195]
```

```
[2.0 2207]
[3.0 2282]
[4.0 2939]
[5.0 1017]
[6.0 56]
[7.0 298]
[8.0 110]
[1.0 2638]
[2.0 2458]
[3.0 3368]
[4.0 1892]
[5.0 9567]
[6.0 183]
[7.0 2571]
[8.0 937]
[9.0 293]
[10.0 3799]
[11.0 577]
[1.0 4112]
[2.0 1221]
[3.0 504]
[4.0 1390]
[1.0 289]
[1.0 98]
[1.0 2069]
[2.0 541]
[2.0 201]
[1.0 81]
[2.0 327]
[1.0 194]
[1.0 1614]
[2.0 254]
[3.0 454]
[4.0 55]
[1 364]
[1 86]
[1 248]
[1 34]
[1 22]
[1.0 404]
[1.0 432]
[1.0 890]
[2.0 230]
[1.0 112]
[1.0 20]
[1.0 46]
[1.0 118]
[1.0 78]
```

```
[1.0 11]
[1.0 2871]
[2.0 1606]
[3.0 1118]
[4.0 1042]
[5.0 2138]
[1.0 1000]
[2.0 348]
[3.0 1652]
[4.0 790]
[5.0 2389]
[6.0 425]
[7.0 3424]
[1.0 2505]
[2.0 1832]
[3.0 675]
[4.0 1240]
[1.0 1405]
[2.0 1115]
[3.0 646]
[4.0 2021]
[5.0 675]
[6.0 492]
[7.0 499]
[8.0 39]
[9.0 138]
[10.0 618]
[1.0 785]
[2.0 430]
[3.0 639]
[4.0 2446]
[5.0 423]
[6.0 172]
[7.0 660]
[8.0 1924]
[9.0 2064]
[1.0 1572]
[2.0 1196]
[3.0 253]
[4.0 1345]
[1.0 1428]
[2.0 433]
[3.0 97]
[1.0 843]
[2.0 1994]
[3.0 533]
[4.0 1908]
[5.0 198]
```

```
[1.0 2347]
[2.0 1591]
[3.0 2555]
[4.0 346]
[5.0 570]
[6.0 655]
[1.0 1126]
[2.0 23]
[3.0 91]
[4.0 510]
[5.0 36]
[1.0 1039]
[2.0 1837]
[3.0 1183]
[1.0 1984]
[2.0 193]
[1.0 28]
[1.0 43]
[1.0 116]
[1.0 1251]
[2.0 2143]
[1.0 77]
[1.0 149]
[1.0 147]
[1.0 82]
[1.0 2634]
[2.0 2528]
[3.0 1327]
[4.0 2162]
[5.0 1130]
[6.0 672]
[7.0 1782]
[8.0 231]
[9.0 143]
[10.0 437]
[1.0 2898]
[2.0 2931]
[3.0 1473]
[4.0 1623]
[5.0 438]
[6.0 168]
[7.0 1669]
[8.0 1579]
[1.0 7170]
[2.0 3099]
[3.0 347]
[4.0 188]
[5.0 2984]
```

```
[6.0 68]
[7.0 937]
[1.0 9925]
[2.0 1464]
[3.0 4320]
[4.0 2391]
[5.0 6372]
[6.0 632]
[7.0 2173]
[8.0 1]
[9.0 27]
[10.0 37]
[1.0 2083]
[2.0 3244]
[3.0 0]
[4.0 638]
[5.0 481]
[6.0 1962]
[7.0 1418]
[8.0 257]
[9.0 1643]
[10.0 1875]
[1.0 5418]
[2.0 2759]
[3.0 2125]
[4.0 1889]
[5.0 460]
[6.0 1472]
[7.0 4550]
[8.0 2110]
[9.0 5134]
[10.0 488]
[11.0 148]
[1.0 89]
[1.0 277]
[1.0 182]
[1.0 52]
[2.0 114]
[3.0 117]
[4.0 10]
[1.0 117]
[1.0 2735]
[2.0 817]
[1.0 113]
[1.0 2152]
[2.0 972]
[3.0 1666]
[4.0 594]
```

```
 [1.0 1010]
 [2.0 3959]
 [3.0 770]
 [4.0 1185]
 [1.0 2543]
 [2.0 370]
 [3.0 815]
 [1.0 3107]
 [2.0 761]
 [3.0 742]
 [4.0 448]
 [5.0 687]
 [1.0 2445]
 [2.0 1332]
 [3.0 1589]
 [4.0 678]
 [5.0 2776]
 [1.0 2445]
 [2.0 1105]
 [1.0 202]
 [2.0 56]
 [1.0 32]
 [1.0 82]
 [2.0 48]
 [3.0 101]
 [1.0 1531]
 [2.0 79]
 [3.0 82]
 [1.0 1669]
 [2.0 125]
 [3.0 2568]
 [4.0 469]]
```

```
[9]: print(y)
```

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

[10]:

```
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=0)
```

[11]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

[12]:

```
print(x_train)
```

```
[[ 2.69369947 -0.34082057]
 [-0.95373712 -0.71640086]
 [ 1.07261654 -0.66397761]
 [ 0.26207507 -0.50383534]
 [-0.95373712 -0.0384892 ]
 [ 2.28842873  0.39525936]
 [-0.14319566 -0.1483626 ]
 [ 1.883158    0.73062456]
 [-0.54846639 -0.7630791 ]
 [-0.95373712 -0.67905827]
 [-0.14319566 -0.65751447]
 [-0.14319566  0.1467875 ]
 [-0.95373712 -0.72573651]
 [-0.54846639 -0.56415799]
 [-0.54846639 -0.08660369]
 [-0.95373712 -0.47439215]
 [ 2.28842873  0.69759073]
 [-0.95373712 -0.29773296]
 [-0.54846639  0.53098532]
 [-0.54846639  1.19668883]
 [ 1.07261654 -0.75446158]
 [-0.95373712 -0.72860902]
 [ 0.66734581  0.93098193]
 [-0.14319566  0.4017225 ]
 [-0.14319566 -0.25177285]
 [-0.95373712  4.79163141]
 [-0.95373712 -0.58785617]
 [-0.54846639 -0.4758284 ]
 [ 2.28842873  2.90223989]
 [ 0.66734581 -0.48013716]
 [-0.95373712 -0.74009905]
 [-0.95373712  0.31482917]
 [-0.95373712 -0.67834015]
 [ 0.26207507  0.10585428]
 [-0.54846639 -0.61945375]
 [ 1.883158   -0.66613199]
 [ 0.66734581 -0.47008339]
```

```
[-0.95373712  2.16831434]
[ 0.66734581  3.79128083]
[-0.54846639 -0.72142775]
[-0.95373712  0.80890038]
[ 0.26207507  0.57407293]
[-0.14319566 -0.69916582]
[-0.95373712  0.70979889]
[ 0.26207507  0.18125759]
[ 2.69369947 -0.58282929]
[-0.14319566 -0.70060208]
[ 0.66734581 -0.29988734]
[ 1.07261654 -0.47941903]
[-0.14319566  0.2056739 ]
[ 2.28842873 -0.76523348]
[ 1.47788727 -0.63884317]
[-0.95373712 -0.75374346]
[ 2.28842873 -0.64243381]
[-0.95373712 -0.4944997 ]
[-0.54846639 -0.66182323]
[ 0.26207507 -0.53615105]
[ 0.26207507  0.05989417]
[-0.95373712 -0.57708427]
[-0.14319566  0.16833131]
[ 0.26207507 -0.41837826]
[-0.14319566 -0.5052716 ]
[-0.95373712 -0.67403139]
[ 0.26207507 -0.46290212]
[ 1.47788727 -0.03633482]
[-0.95373712 -0.65607822]
[-0.54846639 -0.53471479]
[ 0.26207507 -0.61658124]
[ 1.47788727 -0.67044075]
[ 0.26207507 -0.67187701]
[ 0.26207507  3.0580734 ]
[ 0.66734581 -0.75877034]
[-0.95373712  0.70118137]
[-0.54846639 -0.75015282]
[-0.95373712 -0.06649614]
[-0.54846639 -0.64602444]
[ 0.66734581 -0.50239909]
[-0.95373712  1.47675827]
[ 1.07261654 -0.73579029]
[-0.95373712  0.41393066]
[-0.95373712 -0.77672351]
[ 0.26207507 -0.08157681]
[-0.14319566  0.26456029]
[-0.54846639  0.17192194]
[-0.95373712 -0.77026037]
```

```
[-0.95373712 -0.52322477]
[-0.54846639 -0.45356647]
[ 1.47788727 -0.57062113]
[-0.95373712 -0.70060208]
[ 0.66734581 -0.64243381]
[-0.14319566 -0.65105133]
[ 1.47788727  0.77586655]
[ 3.0989702  -0.67834015]
[-0.54846639 -0.58713805]
[-0.14319566  0.45773639]
[-0.95373712 -0.14549009]
[-0.95373712  2.53455899]
[-0.95373712 -0.60652747]
[-0.54846639 -0.23812844]
[-0.95373712 -0.75158908]
[ 0.26207507 -0.45572085]
[ 1.47788727  1.67424313]
[-0.95373712 -0.70419271]
[-0.14319566  1.05019097]
[-0.95373712  0.02398783]
[-0.95373712  1.29650845]
[-0.14319566 -0.29988734]
[-0.95373712 -0.61442686]
[-0.54846639 -0.54979546]
[-0.14319566 -0.7120921 ]
[-0.95373712 -0.70850147]
[-0.95373712 -0.17852392]
[-0.95373712  0.71123514]
[-0.14319566 -0.57923865]
[-0.95373712  0.22434519]
[-0.14319566 -0.25608161]
[ 1.47788727 -0.7523072 ]
[ 1.883158   -0.63668879]
[-0.14319566  0.05199477]
[-0.95373712 -0.76164285]
[ 0.26207507 -0.29773296]
[-0.95373712  3.10618789]
[-0.14319566 -0.70562896]
[-0.14319566 -0.73148153]
[-0.54846639  0.53457596]
[-0.95373712 -0.26757164]
[ 2.28842873 -0.68552141]
[-0.54846639 -0.72789089]
[-0.54846639 -0.21730276]
[-0.95373712 -0.51101661]
[ 0.26207507 -0.35805561]
[-0.95373712 -0.7602066 ]
[-0.95373712  0.97119703]
```

```
[-0.54846639 -0.74440781]
[ 0.26207507 -0.03633482]
[ 0.66734581 -0.52609727]
[-0.95373712  0.25307026]
[ 1.07261654 -0.08085868]
[ 0.66734581 -0.52107039]
[ 1.47788727  1.061681  ]
[-0.54846639 -0.69485706]
[ 0.66734581 -0.45859336]
[ 0.26207507  0.97191515]
[ 1.07261654 -0.65320571]
[-0.14319566 -0.45859336]
[-0.95373712  1.44659694]
[ 1.47788727 -0.5993462 ]
[-0.14319566 -0.20437648]
[-0.54846639  1.32666978]
[-0.95373712 -0.70706522]
[-0.95373712 -0.467929  ]
[-0.14319566 -0.40186134]
[ 1.07261654 -0.43633143]
[ 1.07261654 -0.63884317]
[ 1.07261654 -0.33076679]
[-0.95373712 -0.64530632]
[-0.95373712  4.36434598]
[ 0.66734581 -0.2611085 ]
[-0.95373712  1.01428463]
[ 0.66734581  1.35826735]
[ 1.883158   -0.75661596]
[-0.54846639  1.32020663]
[-0.95373712 -0.53256041]
[-0.54846639  0.36868867]
[ 2.69369947 -0.47080151]
[ 0.26207507  1.01643901]
[ 0.66734581 -0.37529065]
[-0.14319566 -0.7013202 ]
[ 1.883158   -0.61586312]
[-0.54846639  0.98053268]
[ 0.66734581  0.02686033]
[ 0.66734581 -0.29126982]
[ 0.66734581 -0.4542846 ]
[ 1.883158   -0.70562896]
[-0.95373712 -0.74728032]
[-0.54846639 -0.42196889]
[ 1.883158    0.59705298]
[-0.54846639 -0.53615105]
[-0.54846639  1.03080155]
[ 1.883158   -0.73507216]
[-0.14319566 -0.71496461]
```

```
[-0.95373712 -0.33866619]
[-0.14319566  0.35648052]
[-0.95373712 -0.46433837]
[ 0.26207507 -0.69557519]
[ 0.26207507  0.58556296]
[-0.54846639 -0.5993462 ]
[ 0.26207507 -0.74512594]
[-0.14319566 -0.53543292]
[-0.54846639  1.44085193]
[ 2.69369947 -0.75805222]
[-0.95373712 -0.09019433]
[-0.95373712 -0.69988395]
[ 0.66734581 -0.61873563]
[ 1.07261654 -0.30204172]
[ 0.26207507  1.09327858]
[ 0.66734581  6.08569583]
[-0.54846639 -0.65105133]
[-0.54846639 -0.42555953]
[-0.95373712 -0.35805561]
[-0.54846639 -0.39611633]
[ 1.883158   -0.60006433]
[ 0.66734581 -0.56056736]
[-0.95373712 -0.6395613 ]
[ 0.26207507  1.32595165]
[ 1.07261654 -0.6611051 ]
[-0.95373712  1.12272177]
[-0.54846639  2.05844095]
[-0.54846639 -0.11963752]
[ 2.69369947 -0.43417705]
[-0.95373712  2.81031967]
[-0.95373712 -0.71783712]
[ 2.28842873 -0.68193078]
[-0.14319566 -0.48085529]
[-0.54846639 -0.60221871]
[-0.54846639 -0.76810599]
[-0.14319566  0.01824281]
[-0.54846639  0.88071306]
[-0.54846639  2.03115213]
[ 1.07261654 -0.74440781]
[-0.95373712  0.49938774]
[-0.54846639 -0.70275646]
[-0.14319566  1.05952662]
[-0.14319566 -0.60293684]
[-0.14319566  2.31768471]
[-0.95373712 -0.72645464]
[ 1.07261654 -0.70993772]
[ 1.883158    0.82326291]
[ 0.26207507  0.57191855]
```

```
[-0.14319566  0.41177628]
[ 0.66734581 -0.67618577]
[ 0.26207507 -0.32573991]
[-0.95373712 -0.17924205]
[ 1.07261654 -0.63884317]
[-0.95373712  0.11375367]
[-0.14319566 -0.365955  ]
[ 1.47788727 -0.11173813]
[ 1.47788727 -0.31065924]
[-0.95373712 -0.65392384]
[-0.95373712 -0.42268702]
[-0.54846639 -0.33220305]
[-0.95373712 -0.05931488]
[-0.54846639  0.35791677]
[-0.95373712 -0.72070963]
[-0.95373712 -0.59431932]
[-0.54846639 -0.60150058]
[ 1.07261654  0.27245969]
[-0.54846639 -0.65966885]
[-0.95373712 -0.7013202 ]
[-0.95373712 -0.17565141]
[-0.95373712  1.10979549]
[ 1.883158   -0.74871657]
[-0.95373712 -0.75805222]
[ 0.26207507  0.66671128]
[-0.54846639 -0.64027943]
[-0.95373712 -0.69126643]
[-0.95373712 -0.62735315]
[-0.95373712 -0.212994  ]
[-0.14319566  1.63402803]
[-0.95373712  1.04157345]
[-0.95373712 -0.76451536]
[-0.54846639 -0.19791334]
[-0.95373712  0.83188044]
[-0.14319566  0.06492105]
[-0.14319566 -0.2316653 ]
[-0.14319566  0.85414237]
[ 1.07261654 -0.50670785]
[ 1.47788727  0.23368084]
[ 0.26207507 -0.32645803]
[ 0.26207507  0.21357329]
[-0.14319566 -0.32071302]
[-0.54846639 -0.48875468]
[-0.54846639 -0.47367402]
[-0.14319566 -0.47439215]
[-0.54846639  1.54498031]
[-0.54846639 -0.5454867 ]
[-0.54846639  3.76399201]
```

```
 [ 1.07261654  0.6243418 ]
 [ 0.66734581  1.20889699]
 [ 1.07261654 -0.69126643]
 [-0.54846639  0.00890717]
 [-0.54846639 -0.70060208]
 [ 3.0989702  -0.37026377]
 [ 1.883158    0.34929925]
 [-0.54846639 -0.48731843]
 [-0.95373712  0.43403821]
 [-0.95373712 -0.28480668]
 [-0.54846639  0.75432275]
 [-0.54846639  0.01608843]
 [ 1.07261654 -0.31424988]
 [-0.95373712 -0.72286401]
 [-0.14319566 -0.72214588]
 [ 0.26207507 -0.77744164]
 [-0.54846639  0.09220987]
 [-0.95373712 -0.2208934 ]
 [-0.54846639  0.26671467]
 [ 2.69369947  1.94354066]
 [ 0.26207507  2.95969003]
 [-0.54846639 -0.42986829]
 [ 0.26207507  1.33456917]]
```

[13]:

```
print(x_test)
```

```
[[-0.54846639  0.1467875 ]
 [-0.14319566  0.27317781]
 [ 2.69369947  0.56186477]
 [-0.14319566 -0.42268702]
 [ 1.07261654 -0.43130454]
 [-0.14319566 -0.67546764]
 [-0.95373712 -0.74225343]
 [-0.54846639  0.0742567 ]
 [ 1.47788727 -0.76236098]
 [-0.95373712 -0.76882412]
 [-0.95373712  0.58843546]
 [ 0.26207507  1.05378161]
 [ 2.69369947 -0.59575557]
 [ 1.47788727  2.48285386]
 [-0.14319566 -0.65248758]
 [-0.54846639 -0.64458819]
 [-0.95373712 -0.72932715]
 [ 1.07261654 -0.76666974]
 [-0.95373712  0.37443369]
 [-0.95373712 -0.71424648]
 [-0.54846639  0.80028286]
 [-0.54846639  0.64732186]
```

```
[-0.54846639  0.69974511]
[-0.14319566  0.74139646]
[-0.95373712  2.22791886]
[ 1.47788727  0.41393066]
[ 2.28842873 -0.54189606]
[-0.14319566 -0.69270268]
[-0.95373712  0.64014059]
[ 2.28842873 -0.50598972]
[-0.54846639 -0.23741031]
[ 1.47788727  0.49507898]
[-0.14319566 -0.78462291]
[-0.95373712  1.27711903]
[-0.14319566 -0.77169663]
[ 1.47788727 -0.01191851]
[ 0.26207507  0.43691072]
[ 0.66734581  0.85845113]
[ 0.66734581 -0.05428799]
[ 0.26207507 -0.44782146]
[ 1.47788727 -0.37170002]
[-0.95373712  6.34278521]
[-0.14319566 -0.74584406]
[-0.54846639 -0.10742937]
[-0.95373712  1.10692299]
[-0.14319566  0.16904943]
[-0.14319566 -0.1993496 ]
[ 0.66734581 -0.48085529]
[-0.14319566  0.18484822]
[ 0.26207507 -0.21730276]
[ 1.47788727 -0.42627765]
[-0.14319566 -0.65536009]
[-0.95373712 -0.67762202]
[-0.14319566  0.37946058]
[-0.14319566 -0.71927337]
[ 0.26207507 -0.64961508]
[-0.95373712 -0.14477196]
[-0.95373712 -0.26182662]
[-0.95373712 -0.20724899]
[ 0.26207507 -0.61442686]
[ 0.66734581 -0.1806783 ]
[-0.95373712  0.90082061]
[-0.95373712 -0.12179191]
[-0.95373712 -0.58570179]
[ 2.28842873 -0.57421177]
[ 0.26207507  0.06635731]
[ 0.26207507 -0.64387006]
[-0.54846639 -0.56343986]
[-0.95373712 -0.75877034]
[ 0.26207507  0.38089683]
```

```
 [-0.95373712  0.42326631]
 [-0.95373712  0.34427236]
 [-0.95373712 -0.70491084]
 [-0.95373712 -0.08660369]
 [ 0.26207507 -0.57205739]
 [-0.54846639 -0.67834015]
 [-0.95373712 -0.70347458]
 [ 1.883158   -0.61873563]
 [-0.95373712  1.17945379]
 [ 1.883158   -0.78390478]
 [ 0.66734581 -0.43920393]
 [-0.54846639 -0.69772957]
 [-0.95373712  0.143915  ]
 [-0.14319566 -0.4542846 ]
 [ 0.26207507  0.76796715]
 [ 0.66734581  0.75073211]
 [ 2.28842873 -0.69413894]
 [-0.54846639 -0.51891601]
 [-0.14319566 -0.72573651]
 [ 0.26207507  0.93241818]
 [-0.95373712  0.97119703]
 [-0.14319566 -0.32573991]
 [-0.54846639 -0.13687257]
 [-0.54846639 -0.12538254]
 [ 1.883158   -0.11173813]
 [ 0.66734581 -0.37313627]
 [-0.95373712  0.08646486]
 [-0.95373712 -0.72573651]
 [-0.95373712  0.24086211]
 [-0.95373712  0.76078589]]
```

[14]:

```python
print(f"Number of elemen x_train : {len(x_train)}")
print(f"Number of elemen x_test : {len(x_test)}")
print(f"Number of elemen y_train : {len(y_train)}")
print(f"Number of elemen y_train : {len(y_test)}")
```

```
Number of elemen x_train : 300
Number of elemen x_test : 100
Number of elemen y_train : 300
Number of elemen y_train : 100
```

[15]:

```python
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
```

[15]:

```
GaussianNB()
```

[16]:

```python
y_pred = classifier.predict(x_test)
```

[17]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

[18]:

```
print(cm)
```

```
[[88  1]
 [ 7  4]]
```

[19]:

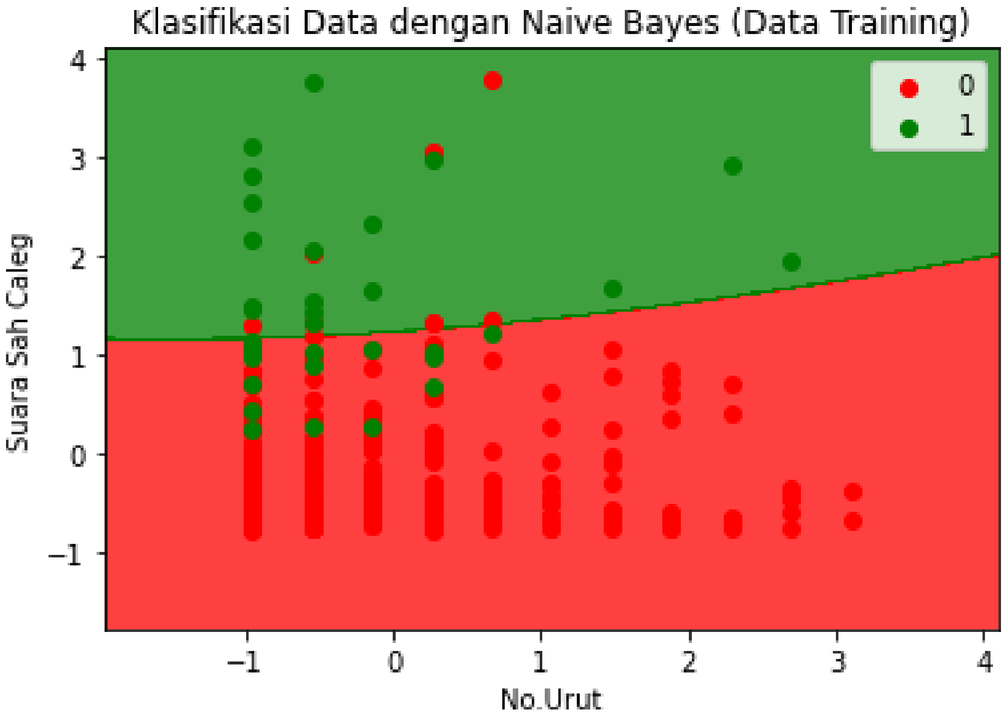
```
from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
```

[20]:

```
x1, x2 = np.meshgrid(
    np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0].max() + 1, step=0.
 ↪01),
    np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 0].max() + 1, step=0.
 ↪01),
)

plt.contourf(x1,x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).
 ↪reshape(x1.shape), alpha=0.75, cmap = ListedColormap(("red", "green")))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i,j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                c = ListedColormap(("red", "green"))(i), label = j)
plt.title("Klasifikasi Data dengan Naive Bayes (Data Training)")
plt.xlabel("No.Urut")
plt.ylabel("Suara Sah Caleg")
plt.legend()
plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```

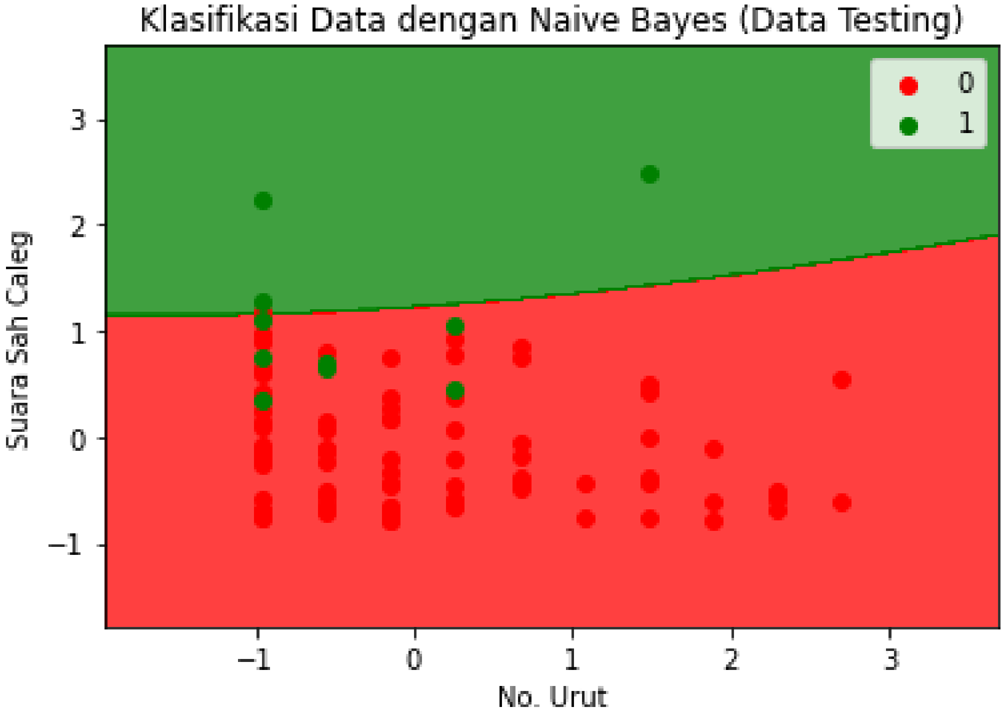


```
[21]: x_set, y_set = x_test, y_test
      x1, x2 = np.meshgrid(
          np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0].max() + 1, step=0.
       ↪01),
          np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 0].max() + 1, step=0.
       ↪01),
      )

      plt.contourf(x1,x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).
       ↪reshape(x1.shape), alpha=0.75, cmap = ListedColormap(("red", "green")))
      plt.xlim(x1.min(), x1.max())
      plt.ylim(x2.min(), x2.max())
      for i,j in enumerate(np.unique(y_set)):
          plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                      c = ListedColormap(("red", "green"))(i), label = j)
      plt.title("Klasifikasi Data dengan Naive Bayes (Data Testing)")
      plt.xlabel("No. Urut")
      plt.ylabel("Suara Sah Caleg")
      plt.legend()
      plt.show()
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
```

```
a single row if you intend to specify the same RGB or RGBA value for all points.
*c* argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
*x* & *y*.  Please use the *color* keyword-argument or provide a 2-D array with
a single row if you intend to specify the same RGB or RGBA value for all points.
```


Klasifikasi Data dengan Naive Bayes (Data Testing)
● 0
● 1
Suara Sah Caleg
No. Urut