TUGAS PEMROGRAMAN 3
PENGANTAR KECERDASAN BUATAN
BINARY CLASSIFICATION MENGGUNAKAN *NAIVE BAYES*

Disusun oleh:

1301218548 – Irgi Ahmad Maulana

1301218586 - Muhamad Azmi Rizkifar

FAKULTAS INFORMATIKA

PROGRAM STUDI S1 INFORMATIKA

UNIVERSITAS TELKOM

2021/2022

Masalah / dataset

Diberikan dataset berupa data yang *continue* dan mahasiswa diminta untuk membuat klasifikasi data yang menghasilkan output berupa 0 atau 1 yang bisa diasumsikan **layak = 1** dan t**idak layak = 0**, klasifikasi ini dinamakan dengan *binary classification.*

Detail proses

Laporan ini memuat proses klasifikasi menggunakan *naive bayes* dengan menggunakan perhitungan peluang Gaussian (Distribusi Normal) karena data yang bersifat *continue*

$$P(x_k|C_i) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x_k-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

1. **Proses download**

   Pada proses ini dilakukan pengunduhan data set menggunakan gdown dan pembacaan dataset yang diberikan dengan menggunakan library pandas.

```
# Melakukan download dataset pada google drive
# https://docs.google.com/spreadsheets/d/187zDetHTNzpwXFC6zfNtKF-GLEjUZtQJ/edit?usp=sharing&ouid=109330686360258100205&rtpof=true&sd=true


!gdown 187zDetHTNzpwXFC6zfNtKF-GLEjUZtQJ
```

```
Downloading...
From: https://drive.google.com/uc?id=187zDetHTNzpwXFC6zfNtKF-GLEjUZtQJ
To: /content/traintest.xlsx
100% 17.7k/17.7k [00:00<00:00, 23.9MB/s]
```

2. **Pembacaaan dan pemisahan data**

   karena data yang diberikan mempunyai *sheet* yang berbeda yaitu data train dan data test maka perlu dilakukan pemisahan data.

```
# Import Library pandas
import pandas as pd

# Proses pembacaan data latih/uji
dataFrameTrain = pd.read_excel('traintest.xlsx')
dataFrameTest = pd.read_excel('traintest.xlsx', sheet_name='test')
dataFrameTrain.head()
```

   Output :

	id	x1	x2	x3	y
0	1	60	64	0	1
1	2	54	60	11	0
2	3	65	62	22	0
3	4	34	60	0	1
4	5	38	69	21	0

3. **Teknik pemrosesan data**

Teknik pra-pemrosesan data dilakukan pada sebuah data sebelum kita melakukan proses selanjutnya, yang kami lakukan terlebih dahulu adalah melakukan pembersihan data, pembersihan data ini masuk ke dalam pra-pemrosesan data. Tujuan dilakukannya pra-pemrosesan data pada sebuah data sebelum melakukan pemrosesan adalah sebagai berikut:

a. Untuk mempermudah memahami data sehingga mempermudah pemilihan teknik dan metode data mining yang tepat.
b. Untuk meningkatkan kualitas data sehingga hasil data mining menjadi lebih baik.
c. Untuk meningkatkan efisiensi dan kemudahan proses penambangan data.

Kami melakukan pengecekkan terlebih dahulu untuk validasi kesesuaian data seperti pada gambar berikut :

```
dataFrameTrain.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 296 entries, 0 to 295
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      296 non-null    int64
 1   x1      296 non-null    int64
 2   x2      296 non-null    int64
 3   x3      296 non-null    int64
 4   y       296 non-null    int64
dtypes: int64(5)
memory usage: 11.7 KB
```

Setelah melakukan pengecekkan data, maka seluruh data yang berjumlah 296 tidak ada yang null yang berarti data sudah sesuai, dan dari setiap data tersebut sudah sesuai dengan tipe datanya yaitu int. Sehingga tidak perlu dilakukan pra-pemrosesan data lebih lanjut.

4. **Pengelompokan data dengan nilai kolom y = 1**

Dilakukan proses perhitungan untuk mendapatkan nilai *mean* dan *standar deviation* dari masing masing kelompok data yang dikelompokkan oleh kolom y dengan nilai y = 1 dan y = 0

```
# describing data to get mean and std
# output data 0 or 1
def att_desc(output_data):
    return dataFrameTrain[dataFrameTrain['y'] ==
output_data].describe()
```

```
# get mean and std for worthy or acceptable
x1meanWorthy = att_desc(1)['x1']['mean']
x1stdWorthy = att_desc(1)['x1']['std']

x2meanWorthy = att_desc(1)['x2']['mean']
x2stdWorthy = att_desc(1)['x2']['std']

x3meanWorthy = att_desc(1)['x3']['mean']
x3stdWorthy = att_desc(1)['x3']['std']

print(x1meanWorthy, x2meanWorthy, x3meanWorthy)
print(x1stdWorthy, x2stdWorthy, x3stdWorthy)
```

Output :

```
51.93577981651376 62.92660550458716 2.8394495412844036
11.110484299554663 3.222062258631095 5.953499042384458
```

5. **Pengelompokan data dengan nilai kolom y = 0**

```
# get mean and std for unworthy or unacceptable
x1meanUnWorthy = att_desc(0)['x1']['mean']
x1stdUnWorthy = att_desc(0)['x1']['std']

x2meanUnWorthy = att_desc(0)['x2']['mean']
x2stdUnWorthy = att_desc(0)['x2']['std']

x3meanUnWorthy = att_desc(0)['x3']['mean']
x3stdUnWorthy = att_desc(0)['x3']['std']
print(x1meanUnWorthy, x2meanUnWorthy, x3meanUnWorthy)
print(x1stdUnWorthy, x2stdUnWorthy, x3stdUnWorthy)
```

Output :

```
53.93589743589744 62.756410256410255 7.666666666666667
10.198471675149396 3.2839076241650407 9.296929623961898
```

6. **Mendapatkan nilai jumlah data yang di filter berdasarkan nilai y = 0 dan y = 1**

```
lenghtOfAcceptableData = len(dataFrameTrain[dataFrameTrain['y'] ==
1.0])
lenghtOfUnAcceptableData = len(dataFrameTrain[dataFrameTrain['y'] ==
0.0])

lenghtOfAllAdata = len(dataFrameTrain)
```

```
print("Acceptable data = {} Unacceptable data = {} All data =
{}".format(lenghtOfAcceptableData, lenghtOfUnAcceptableData,
lenghtOfAllAdata))
```

Output :

```
Acceptable data = 218 Unacceptable data = 78 All data = 296
```

7. **Melakukan proses perhitungan dengan metode gaussian**

```
import math

# calculate probability using formula
def calc_probability(x, mean, std):
  exponent = math.exp(-((x-mean)**2 / (2*std**2)))
  # return (1 / (std * (math.sqrt(2*math.pi)))) * exponent
  return (1 / (math.sqrt(2 * math.pi) * std)) * exponent
```

Setelah itu dibuat fungsi untuk melakukan klasifikasi dengan membandingkan probabilitas yang dikelompokkan oleh nilai y, model didapatkan dari perhitungan calc_probability

```
# decide classification
def decide_classification(x1Test, x2Test, x3Test):
  probWorthy1 = calc_probability(x1Test, x1meanWorthy, x1stdWorthy)
  probWorthy2 = calc_probability(x2Test, x2meanWorthy, x2stdWorthy)
  probWorthy3 = calc_probability(x3Test, x3meanWorthy, x3stdWorthy)
  pWorthy = (lenghtOfAcceptableData/lenghtOfAllAdata) * probWorthy1 *
probWorthy2 * probWorthy3

  probUnWorthy1 = calc_probability(x1Test, x1meanUnWorthy,
x1stdUnWorthy)
  probUnWorthy2 = calc_probability(x2Test, x2meanUnWorthy,
x2stdUnWorthy)
  probUnWorthy3 = calc_probability(x3Test, x3meanUnWorthy,
x3stdUnWorthy)
  pUnWorthy = (lenghtOfAcceptableData/lenghtOfAllAdata) *
probUnWorthy1 * probUnWorthy2 * probUnWorthy3

  print('probability for y = 1 -> {}'.format(pWorthy))
  print('probability for y = 0 -> {}'.format(pUnWorthy))

  acceptable = 0
  if(pWorthy > pUnWorthy):
    # test data acceptable
    acceptable = 1
```

```
  else:
    # test data unacceptable
    acceptable = 0
  return acceptable
```

8. **Validasi model dengan data latih (actual data)**

Untuk memastikan kebenaran algoritma dan perhitungan, kami melakukan validasi model dengan *data training* atau *actual data*. Disini kami mengambil sampel data dengan ***id = 1*** dan ***id = 2*** yang memiliki nilai ***y*** yang berbeda.

Validasi model dengan data train (actual data)

id	x1	x2	x3	y
1	60	64	0	1
2	54	60	11	0

Perhitungan terbukti benar, output **y** sesuai dengan *train data* atau *actual data*.

```
# Contoh pengujian model dari data train dengan y = 1
print('====== Contoh pengujian model dari data train dengan y = 1 ======')
example1 = decide_classification(60, 64, 0)
print('x1 x2 x3 -> y')
print('60 64 0  -> {}\n'.format(example1))

# Contoh pengujian model dari data train dengan y = 0
print('====== Contoh pengujian model dari data train dengan y = 0 ======')
example2 = decide_classification(54, 60, 11)
print('x1 x2 x3 -> y')
print('54 60 11 -> {}'.format(example2))
```

```
====== Contoh pengujian model dari data train dengan y = 1 ======
probability for y = 1 -> 0.00014235259904175774
probability for y = 0 -> 8.33768749286531e-05
x1 x2 x3 -> y
60 64 0  -> 1

====== Contoh pengujian model dari data train dengan y = 0 ======
probability for y = 1 -> 5.579842149789035e-05
probability for y = 0 -> 9.901868657641887e-05
x1 x2 x3 -> y
54 60 11 -> 0
```

Output pelatihan(training) dan pengujian (testing)

```
Hasil prediksi :
probability for y = 1 -> 7.481434681154435e-05
probability for y = 0 -> 3.648713421898175e-05
43 59 2  -> 1

probability for y = 1 -> 4.955626417659501e-05
probability for y = 0 -> 2.889284011041653e-05
67 66 0  -> 1

probability for y = 1 -> 0.00012510034528011684
probability for y = 0 -> 8.598921339659057e-05
58 60 3  -> 1

probability for y = 1 -> 0.00021175024633781784
probability for y = 0 -> 0.00011745111797483866
49 63 3  -> 1

probability for y = 1 -> 0.00010668196392365885
probability for y = 0 -> 5.1199084502255435e-05
45 60 0  -> 1

probability for y = 1 -> 6.387932395668812e-05
probability for y = 0 -> 4.0683407815192054e-05
54 58 1  -> 1

probability for y = 1 -> 0.00013015787337867786
probability for y = 0 -> 7.964777600385363e-05
56 66 3  -> 1

probability for y = 1 -> 2.3732606847822945e-05
probability for y = 0 -> 9.606952640595338e-06
42 69 1  -> 1

probability for y = 1 -> 0.00010182474623897358
probability for y = 0 -> 6.0184229176788236e-05
50 59 2  -> 1

probability for y = 1 -> 0.00010590778028007192
probability for y = 0 -> 6.6440121837383e-05
59 60 0  -> 1
```

Hasil *file* excell untuk *data test*

	A	B	C	D	E	F
1		**id**	**x1**	**x2**	**x3**	**y**
2	**0**	297	43	59	2	1
3	**1**	298	67	66	0	1
4	**2**	299	58	60	3	1
5	**3**	300	49	63	3	1
6	**4**	301	45	60	0	1
7	**5**	302	54	58	1	1
8	**6**	303	56	66	3	1
9	**7**	304	42	69	1	1
10	**8**	305	50	59	2	1
11	**9**	306	59	60	0	1