

<<<<

# ARTIFICIAL INTELLIGENCE (AI)

## PRAKTIKUM 11

Dosen Pengampu : Renovita Edelani S.ST., M.Tr.Kom

---

oleh : Dukhaan Kamimpangan  
2 D4 IT A  
3122600003



01.

DATATRaining  
MILK\_TRAINING.CSV

---



# MENAMPILKAN DATATRAINING MILK\_TRAINING.CSV

Analisis : Membaca data training dari file CSV dan memisahkan fitur (pH, Temperatur, Rasa, Bau, Lemak, Kekeruhan, Warna) dari label kelas (Grade) dengan menggunakan Pandas. Data training terdiri dari 742 sampel, di mana setiap sampel memiliki 7 fitur yang berbeda. Label kelas menunjukkan kategori kualitas susu (misalnya, "tinggi", "sedang", "rendah").

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score

# 1. Membaca data training
data_training = pd.read_csv("milk_training.csv")
train_data = data_training.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
train_label = data_training.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

print(train_data)
print(train_label)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour
0	6.6	36	1	0	1	0	254
1	6.6	36	0	1	0	1	253
2	6.6	37	1	1	1	1	255
3	6.6	45	0	1	1	1	255
4	6.6	45	0	1	1	1	250
...	...	...	...	...	...	...	...
727	6.7	45	1	1	1	0	245
738	6.5	38	1	0	1	0	255
739	6.7	41	1	0	0	0	247
740	6.5	41	0	0	0	0	255
741	6.8	38	0	0	0	0	255

```
[742 rows x 7 columns]
0      high
1      high
2      high
3      high
4      high
...
727    medium
738    medium
739    medium
740    medium
741    medium
Name: Grade, Length: 742, dtype: object
```

/[AI]

02.

DATATEST

MILK\_TESTING.CSV

---

# MENAMPILKAN DATATEST MILK\_TESTING.CSV

Analisis :

berisi informasi tentang susu. Data training terdiri dari 742 sampel, dengan setiap sampel memiliki 7 fitur yang berbeda (pH, Temperatur, Rasa, Bau, Lemak, Kekeruhan, dan Warna). Selain itu, setiap sampel memiliki label kelas yang menunjukkan kualitas susu ("tinggi", "sedang", atau "rendah").

```
data_test = pd.read_csv("milk_testing.csv")
test_data = data_test.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
test_label = data_test.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

print(train_data)
print(train_label)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour
0	6.6	35	1	0	1	0	254
1	6.6	36	0	1	0	1	253
2	6.6	37	1	1	1	1	255
3	6.8	45	0	1	1	1	255
4	6.6	45	0	1	1	1	250
...	...	...	...	...	...	...	...
737	6.7	45	1	1	1	0	245
738	6.5	38	1	0	1	0	255
739	6.7	41	1	0	0	0	247
740	6.8	41	0	0	0	0	255
741	6.8	38	0	0	0	0	255

```
[742 rows x 7 columns]
0      high
1      high
2      high
3      high
4      high
...
737   medium
738   medium
739   medium
740   medium
741   medium
Name: Grade, Length: 742, dtype: object
```

**03.**

## **KLASIFIKASI NAÏVE BAYES**

---

# MENAMPILKAN DATATEST MILK\_TESTING.CSV

Analisis :

Hasil akurasi menunjukkan bahwa:

- Akurasi Gaussian Naive Bayes: sekitar 95.58%
- Akurasi Multinomial Naive Bayes: sekitar 58.99%
- Akurasi Bernoulli Naive Bayes: sekitar 62.78%

Dari hasil tersebut, terlihat bahwa model Gaussian Naive Bayes memiliki akurasi yang paling tinggi diikuti oleh Bernoulli Naive Bayes dan Multinomial Naive Bayes. Ini menunjukkan bahwa distribusi fitur dalam dataset mungkin lebih cocok dengan asumsi distribusi normal (Gaussian) dari model Gaussian Naive Bayes.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score

# 1. Memuat data training
data_training = pd.read_csv("milk_training.csv")
train_data = data_training.dropna().drop("grade", axis=1) # menghapus baris yang memiliki nilai nan dan kolom "grade"
train_label = data_training.dropna()["grade"] # mengambil kolom "grade" sebagai label

# 2. Memuat data testing
data_test = pd.read_csv("milk_testing.csv")
test_data = data_test.dropna().drop("grade", axis=1) # menghapus baris yang memiliki nilai nan dan kolom "grade"
test_label = data_test.dropna()["grade"] # mengambil kolom "grade" sebagai label

# 3. Inisialisasi dan pelatihan model Gaussian Naive Bayes
gnb = GaussianNB()
gnb.fit(train_data, train_label)

# Prediksi dengan Gaussian Naive Bayes pada data testing
gnb_predictions = gnb.predict(test_data)
gnb_accuracy = accuracy_score(test_label, gnb_predictions)
print("Gaussian Naive Bayes Accuracy:", gnb_accuracy)

# 4. Inisialisasi dan pelatihan model Multinomial Naive Bayes
mnb = MultinomialNB()
mnb.fit(train_data, train_label)

# Prediksi dengan Multinomial Naive Bayes pada data testing
mnb_predictions = mnb.predict(test_data)
mnb_accuracy = accuracy_score(test_label, mnb_predictions)
print("Multinomial Naive Bayes Accuracy:", mnb_accuracy)

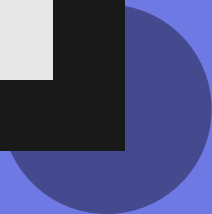
# 5. Inisialisasi dan pelatihan model Bernoulli Naive Bayes
bnb = BernoulliNB()
bnb.fit(train_data, train_label)

# Prediksi dengan Bernoulli Naive Bayes pada data testing
bnb_predictions = bnb.predict(test_data)
bnb_accuracy = accuracy_score(test_label, bnb_predictions)
print("Bernoulli Naive Bayes Accuracy:", bnb_accuracy)
```

Gaussian Naive Bayes Accuracy: 0.955835962451485  
Multinomial Naive Bayes Accuracy: 0.589904817708152  
Bernoulli Naive Bayes Accuracy: 0.6277682523650986

04.

# NORMALISASI DATA TRAINING





# NORMALISASI DATA TRAINING

Analisis :

pada baris pertama hasil normalisasi, nilai pH dinormalisasi menjadi sekitar 0.55, nilai Temperatur menjadi sekitar 0.018, Taste memiliki nilai 1.0 (tidak berubah karena sudah dalam rentang 0-1), dan seterusnya

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score

# 1. Membaca data training
data_training = pd.read_csv("milk_training.csv")
train_data = data_training.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
train_label = data_training.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

# 2. Membaca data testing
data_test = pd.read_csv("milk_testing.csv")
test_data = data_test.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
test_label = data_test.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

# 3. Normalisasi menggunakan metode min-max pada data training
max_element_column = np.max(train_data, 0)
min_element_column = np.min(train_data, 0)
newmax = 0
newmin = 1
train_data_scaled = ((train_data - min_element_column)*(newmax-newmin)/(max_element_column-min_element_column))+newmin
print(train_data_scaled)
```

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour
0	0.553846	0.017857	1.0	0.0	1.0	0.0	0.933333
1	0.553846	0.035714	0.0	1.0	0.0	1.0	0.866667
2	0.553846	0.053571	1.0	1.0	1.0	1.0	1.000000
3	0.584615	0.196429	0.0	1.0	1.0	1.0	1.000000
4	0.553846	0.196429	0.0	1.0	1.0	1.0	0.666667
...	...	...	...	...	...	...	...
737	0.569231	0.196429	1.0	1.0	1.0	0.0	0.333333
738	0.538462	0.071429	1.0	0.0	1.0	0.0	1.000000
739	0.569231	0.125000	1.0	0.0	0.0	0.0	0.466667
740	0.584615	0.125000	0.0	0.0	0.0	0.0	1.000000
741	0.584615	0.071429	0.0	0.0	0.0	0.0	1.000000

[742 rows x 7 columns]



**05.**



# **KLASIFIKASI NORMALISASI & NON NORMALISASI**

---

# KLASIFIKASI NORMALISASI & NON NORMALISASI

## Analisis :

melakukan normalisasi menggunakan metode min-max pada data training dan data test, kemudian melakukan klasifikasi menggunakan tiga metode Naive Bayes yang berbeda: Gaussian Naive Bayes, Multinomial Naive Bayes, dan Bernoulli Naive Bayes.

- Pertama, data training dibaca dari file "milk\_training.csv", dan fitur dan label dipisahkan.
- Kemudian, data testing dibaca dari file "milk\_testing.csv", dan juga fitur dan label dipisahkan.
- Proses normalisasi dilakukan menggunakan metode min-max pada data training, lalu nilai minimum dan maksimum dari data training digunakan untuk melakukan normalisasi pada data testing.
- Selanjutnya, tiga model klasifikasi Naive Bayes (Gaussian, Multinomial, dan Bernoulli) dilatih menggunakan data training yang sudah dinormalisasi.
- Setelah pelatihan, prediksi dilakukan pada data testing untuk setiap metode Naive Bayes, dan hasil prediksi dicetak.

```
[15]: import pandas as pd
import numpy as np
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score

# 1. Membaca data training
data_training = pd.read_csv("milk_training.csv")
train_data = data_training.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
train_label = data_training.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

# 2. Membaca data testing
data_test = pd.read_csv("milk_testing.csv")
test_data = data_test.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
test_label = data_test.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

# 3. Normalisasi menggunakan metode min-max pada data training
max_element_column = np.max(train_data, 0)
min_element_column = np.min(train_data, 0)
newmin = 0
newmax = 1
train_data_scaled = ((train_data - min_element_column) * (newmax - newmin) / (max_element_column - min_element_column)) + newmin

# Normalisasi menggunakan metode min-max pada data test dengan menggunakan nilai min dan max dari data training
test_data_scaled = ((test_data - min_element_column) * (newmax - newmin) / (max_element_column - min_element_column)) + newmin

# 4. Klasifikasi menggunakan Gaussian Naive Bayes
gnb = GaussianNB()
gnb.fit(train_data, train_label)
gnb_result = gnb.predict(test_data)
print(gnb_result)

# 5. Klasifikasi menggunakan Multinomial Naive Bayes
mnb = MultinomialNB()
mnb.fit(train_data, train_label)
mnb_result = mnb.predict(test_data)
print(mnb_result)

# 6. Klasifikasi menggunakan Bernoulli Naive Bayes
bnb = BernoulliNB()
bnb.fit(train_data, train_label)
bnb_result = bnb.predict(test_data)
print(bnb_result)
```

ARTI  
SIAL  
TE  
EN  
(AI)[illegible]

06.



## PERBANDINGAN NILAI AKURASI KLASIFIKASI

---

# PERBANDINGAN NILAI AKURASI KLASIFIKASI

## Analisis :

Hasilnya menunjukkan bahwa akurasi model Gaussian Naive Bayes tidak berubah setelah normalisasi. Hal ini mungkin disebabkan oleh sifat alami dari algoritma Gaussian Naive Bayes yang tidak sensitif terhadap skala data. Meskipun normalisasi tidak memengaruhi kinerja model Gaussian Naive Bayes dalam hal akurasi, tetap penting untuk melakukannya karena beberapa algoritma machine learning lainnya mungkin akan terpengaruh oleh skala data. Dengan normalisasi, kita memastikan bahwa data memiliki rentang nilai yang seragam

```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler

# 1. Membaca data training
data_training = pd.read_csv("milk_training.csv")
train_data = data_training.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
train_label = data_training.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

# 2. Membaca data testing
data_test = pd.read_csv("milk_testing.csv")
test_data = data_test.dropna().drop("Grade", axis=1) # Menghapus baris yang memiliki nilai NaN dan kolom "Grade"
test_label = data_test.dropna()["Grade"] # Mengambil kolom "Grade" sebagai label

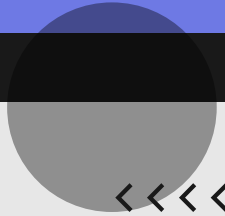
# 3. Normalisasi menggunakan metode min-max pada data training
scaler = MinMaxScaler()
train_data_normalized = scaler.fit_transform(train_data)
test_data_normalized = scaler.transform(test_data)

# 4. Klasifikasi menggunakan Gaussian Naive Bayes tanpa normalisasi
gnb = GaussianNB()
gnb.fit(train_data, train_label)
gnb_predictions = gnb.predict(test_data)
gnb_accuracy = accuracy_score(test_label, gnb_predictions)

# 5. Klasifikasi menggunakan Gaussian Naive Bayes dengan normalisasi
gnb_normalized = GaussianNB()
gnb_normalized.fit(train_data_normalized, train_label)
gnb_predictions_normalized = gnb_normalized.predict(test_data_normalized)
gnb_accuracy_normalized = accuracy_score(test_label, gnb_predictions_normalized)

# 6. Tampilkan hasil
print("Gaussian Naive Bayes Accuracy (Tanpa Normalisasi):", gnb_accuracy)
print("Gaussian Naive Bayes Accuracy (Dengan Normalisasi):", gnb_accuracy_normalized)
```

Gaussian Naive Bayes Accuracy (Tanpa Normalisasi): 0.9558359621451105  
Gaussian Naive Bayes Accuracy (Dengan Normalisasi): 0.9558359621451105



07.

**METODE TERBAIK**

---



# PENJELASAN METODE TERBAIK

Untuk menentukan metode yang paling baik (dengan akurasi tertinggi) untuk dataset susu ("milk"), kita perlu membandingkan akurasi dari tiga metode klasifikasi Naive Bayes yang berbeda: Gaussian Naive Bayes, Multinomial Naive Bayes, dan Bernoulli Naive Bayes.

Setiap metode Naive Bayes memiliki asumsi yang berbeda tentang distribusi data dan cocok untuk jenis data yang berbeda. Berikut adalah gambaran umum:

1. **Gaussian Naive Bayes** : cocok untuk data yang terdistribusi secara normal (Gaussian) atau hampir Gaussian. Ini berarti bahwa jika data memiliki distribusi yang kontinu, maka metode ini mungkin lebih cocok.
2. **Multinomial Naive Bayes** : cocok untuk data yang memiliki fitur diskrit (misalnya, hitungan kata dalam dokumen). Ini biasanya digunakan dalam klasifikasi teks atau data yang diwakili sebagai hitungan frekuensi.
3. **Bernoulli Naive Bayes** : juga cocok untuk data biner atau data yang diwakili sebagai variabel biner (misalnya, 0 atau 1). Ini sering digunakan dalam klasifikasi teks di mana setiap fitur mewakili keberadaan atau ketiadaan suatu kata dalam dokumen.

Untuk dataset susu, hasil yang paling baik (akurasi tertinggi) bervariasi tergantung pada **sifat-sifat fitur dalam data**, seperti **distribusi, diskritisasi, atau keberadaan variabel biner**. Untuk mengetahui metode mana yang paling cocok, kita perlu mencoba dan membandingkan kinerja ketiga metode tersebut dengan menggunakan teknik **validasi silang dan evaluasi** yang cermat terhadap setiap metode. Dengan demikian, kita bisa mendapatkan pemahaman yang lebih baik tentang metode mana yang memberikan hasil yang terbaik untuk **dataset susu yang spesifik**.



/[AI]/[AI]/



**TERIMA  
KASIH**

ARTI  
CIAL  
INTE  
IGEN  
[AI]