

PRAKTIKUM KE-2
MACHINE LEARNING
ALGORITMA BAYESIAN

Dosen Pengampu : Nur Rosyid Muhtadai S.Kom., M.T



Disusun Oleh:

Muhammad Krisnanda Vilovan Saputra

(3323600010)

Sains Data Terapan A

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
PROGRAM STUDI SAINS DATA TERAPAN

Nomor 3.

Input

```
# Data baru
data = [
    ('Panjang', 'Kecil', 'Ringan', 'Ikan Lele'),
    ('Panjang', 'Sedang', 'Berat', 'Ikan Lele'),
    ('Panjang', 'Lebar', 'Berat', 'Gurami'),
    ('Sedang', 'Kecil', 'Berat', 'Ikan Lele'),
    ('Sedang', 'Sedang', 'Ringan', 'Ikan Lele'),
    ('Sedang', 'Lebar', 'Berat', 'Gurami'),
    ('Kecil', 'Kecil', 'Ringan', 'Gurami'),
    ('Kecil', 'Sedang', 'Ringan', 'Gurami'),
    ('Kecil', 'Lebar', 'Berat', 'Gurami'),
]

# Menghitung total
total_data = len(data)
total_ikan_lele = sum(1 for _, _, _, jenis in data if jenis == "Ikan Lele")
total_ikan_gurami = sum(1 for _, _, _, jenis in data if jenis == "Gurami")

# Probabilitas A Priori
P_lele = total_ikan_lele / total_data
P_gurami = total_ikan_gurami / total_data

# Fungsi untuk menghitung probabilitas kondisional
def calculate_conditional_probability(feature1, feature2, label):
    count = sum(1 for item in data if item[3] == label and item[0] == feature1 and item[1] == feature2)
    total_label = sum(1 for item in data if item[3] == label)
    return count / total_label if total_label > 0 else 0

# Fitur yang ingin diuji
test_feature1 = 'Panjang'
test_feature2 = 'Lebar' # Menggunakan 'Lebar' sesuai dengan data

# Menghitung probabilitas untuk Ikan Lele
P_lele_given_features = P_lele * calculate_conditional_probability(test_feature1, test_feature2, "Ikan Lele")

# Menghitung probabilitas untuk Gurami
P_gurami_given_features = P_gurami * calculate_conditional_probability(test_feature1, test_feature2, "Gurami")

# Menampilkan hasil
print(f"P(Jenis Ikan = Ikan Lele | Fitur) = {P_lele_given_features:.4f}")
print(f"P(Jenis Ikan = Gurami | Fitur) = {P_gurami_given_features:.4f}")

# Normalisasi
total = P_lele_given_features + P_gurami_given_features

if total > 0: # Mencegah pembagian dengan nol
    P_lele_normalized = P_lele_given_features / total
    P_gurami_normalized = P_gurami_given_features / total
else:
    P_lele_normalized = 0
    P_gurami_normalized = 0

# Menampilkan hasil akhir
print(f"Probabilitas akhir:")
print(f"P(Jenis Ikan = Ikan Lele | Fitur) = {P_lele_normalized:.4f}")
print(f"P(Jenis Ikan = Gurami | Fitur) = {P_gurami_normalized:.4f}")

if P_lele_normalized > P_gurami_normalized:
    print("Ikan tersebut termasuk Ikan Lele.")
else:
    print("Ikan tersebut termasuk Gurami.")
```

Output

```
P(Jenis Ikan = Ikan Lele | Fitur) = 0.0000
P(Jenis Ikan = Gurami | Fitur) = 0.1111
Probabilitas akhir:
P(Jenis Ikan = Ikan Lele | Fitur) = 0.0000
P(Jenis Ikan = Gurami | Fitur) = 1.0000
Ikan tersebut termasuk Gurami.
```

Analisis

Analisis Program :

- ◆ Program ini menggunakan metode Naïve Bayes untuk mengklasifikasikan jenis ikan berdasarkan fitur yang diberikan, seperti panjang, lebar, dan berat. Dimulai dengan mendefinisikan data yang berisi informasi tentang ikan, program menghitung total entri serta jumlah setiap jenis ikan untuk menentukan probabilitas A priori. Fungsi yang terpisah digunakan untuk menghitung probabilitas kondisional berdasarkan dua fitur yang diuji. Setelah menghitung probabilitas untuk masing-masing ikan, hasilnya ditampilkan dan dinormalisasi agar bisa dibandingkan. Akhirnya, program menentukan jenis ikan yang lebih mungkin berdasarkan probabilitas yang dinormalisasi, memberikan output yang jelas tentang klasifikasi ikan tersebut.
- ◆ Dapat disimpulkan bahwa program ini secara efektif mengimplementasikan algoritma klasifikasi Naïve Bayes untuk menentukan jenis ikan berdasarkan fitur-fitur tertentu. Dengan menghitung probabilitas A priori dan probabilitas kondisional, program ini mampu memberikan prediksi yang informatif tentang klasifikasi ikan, yang berguna dalam konteks pengolahan data dan machine learning.

Analisis Output :

- ◆ Probabilitas 0.0000 untuk Ikan Lele menunjukkan bahwa tidak ada data atau fitur yang mendukung bahwa ikan tersebut adalah Ikan Lele. Ini bisa disebabkan oleh tidak adanya entri dalam data yang memenuhi kriteria untuk Ikan Lele dengan fitur yang diuji.
- ◆ Probabilitas 0.1111 untuk Gurami menunjukkan bahwa meskipun ada beberapa kemungkinan, nilai ini cukup rendah, tetapi setelah proses normalisasi, probabilitasnya menjadi 1.0000. Ini menunjukkan bahwa dalam konteks pengambilan keputusan, tidak ada alternatif lain yang mungkin lebih kuat, sehingga model sangat yakin bahwa ikan yang dianalisis adalah Gurami.

Nomor 4.

Input

```
# Data
data = [
    (30, 'Tinggi', 'Single', 'Punya', 'Tidak'),
    (30, 'Tinggi', 'Single', 'Tidak', 'Tidak'),
    (35, 'Tinggi', 'Single', 'Punya', 'Iya'),
    (45, 'Sedang', 'Single', 'Punya', 'Iya'),
    (45, 'Rendah', 'Menikah', 'Punya', 'Iya'),
    (45, 'Rendah', 'Menikah', 'Tidak', 'Tidak'),
    (35, 'Rendah', 'Menikah', 'Punya', 'Iya'),
    (30, 'Sedang', 'Single', 'Punya', 'Tidak'),
    (45, 'Sedang', 'Menikah', 'Punya', 'Iya'),
    (30, 'Rendah', 'Menikah', 'Punya', 'Iya'),
    (35, 'Sedang', 'Menikah', 'Punya', 'Iya'),
    (35, 'Tinggi', 'Menikah', 'Punya', 'Iya'),
    (45, 'Sedang', 'Single', 'Tidak', 'Tidak'),
]

# Menghitung total
total_data = len(data)
total_beli_iya = sum(1 for _, _, _, _, beli in data if beli == "Iya")
total_beli_tidak = sum(1 for _, _, _, _, beli in data if beli == "Tidak")

# Probabilitas A Priori
P_iya = total_beli_iya / total_data
P_tidak = total_beli_tidak / total_data

# Fungsi untuk menghitung probabilitas kondisional
def calculate_conditional_probabilities(feature_index, feature_value, label):
    count = sum(1 for item in data if item[-1] == label and item[feature_index] == feature_value)
    total_label = sum(1 for item in data if item[-1] == label)
    return count / total_label if total_label > 0 else 0

# Data kondisi yang ingin diuji
umur = 30
gaji = 'Sedang'
status = 'Menikah'
```

```

# Menghitung probabilitas untuk membeli laptop
P_iya_given_conditions = P_iya
P_tidak_given_conditions = P_tidak

# Menghitung probabilitas kondisional untuk setiap fitur
P_iya_given_conditions *= calculate_conditional_probabilities(0, umur, "Iya") # Umur
P_iya_given_conditions *= calculate_conditional_probabilities(1, gaji, "Iya") # Gaji
P_iya_given_conditions *= calculate_conditional_probabilities(2, status, "Iya") # Status
P_iya_given_conditions *= calculate_conditional_probabilities(3, hutang, "Iya") # Hutang

P_tidak_given_conditions *= calculate_conditional_probabilities(0, umur, "Tidak") # Umur
P_tidak_given_conditions *= calculate_conditional_probabilities(1, gaji, "Tidak") # Gaji
P_tidak_given_conditions *= calculate_conditional_probabilities(2, status, "Tidak") # Status
P_tidak_given_conditions *= calculate_conditional_probabilities(3, hutang, "Tidak") # Hutang

# Menampilkan hasil
print(f"P(Beli Laptop = Iya | X) = {P_iya_given_conditions:.4f}")
print(f"P(Beli Laptop = Tidak | X) = {P_tidak_given_conditions:.4f}")

# Normalisasi
total = P_iya_given_conditions + P_tidak_given_conditions

if total > 0: # Mencegah pembagian dengan nol
    P_iya_normalized = P_iya_given_conditions / total
    P_tidak_normalized = P_tidak_given_conditions / total
else:
    P_iya_normalized = 0
    P_tidak_normalized = 0

# Menampilkan hasil akhir
print(f"Probabilitas akhir:")
print(f"P(Beli Laptop = Iya | X) = {P_iya_normalized:.4f}")
print(f"P(Beli Laptop = Tidak | X) = {P_tidak_normalized:.4f}")

if P_iya_normalized > P_tidak_normalized:
    print("Kemungkinan orang tersebut beli laptop.")
else:
    print("Kemungkinan orang tersebut tidak beli laptop.")

```

Output

```

P(Beli Laptop = Iya | X) = 0.0216
P(Beli Laptop = Tidak | X) = 0.0074
Probabilitas akhir:
P(Beli Laptop = Iya | X) = 0.7455
P(Beli Laptop = Tidak | X) = 0.2545
Kemungkinan orang tersebut beli laptop.

```

Analisis

Analisis Program :

- ◆ Program ini menggunakan metode Naïve Bayes untuk mengklasifikasikan kemungkinan seseorang membeli laptop berdasarkan fitur-fitur tertentu, seperti umur, gaji, status, dan hutang. Dimulai dengan mendefinisikan data, program menghitung total entri dan jumlah orang yang membeli serta tidak membeli laptop untuk mendapatkan probabilitas a priori. Fungsi terpisah digunakan untuk menghitung probabilitas kondisional berdasarkan fitur yang diberikan. Setelah menghitung probabilitas untuk membeli dan tidak membeli laptop dengan mengalikan nilai probabilitas awal dan probabilitas kondisional, hasilnya ditampilkan. Program kemudian dinormalisasi untuk mendapatkan nilai yang

proporsional dan menyimpulkan keputusan akhir tentang kemungkinan individu tersebut membeli laptop.

- ◆ Dapat disimpulkan bahwa program ini efektif dalam menerapkan algoritma Naïve Bayes untuk mengklasifikasikan keputusan membeli laptop berdasarkan informasi demografis dan finansial individu. Dengan menghitung probabilitas a priori dan probabilitas kondisional, program ini dapat memberikan prediksi yang akurat dan berguna, yang dapat diterapkan dalam berbagai konteks, seperti pemasaran dan analisis perilaku konsumen.

Analisis Output :

- ◆ Meskipun probabilitas awal untuk membeli dan tidak membeli laptop tampak rendah, hasil akhir setelah normalisasi menunjukkan bahwa model memproses informasi dengan cara yang mengarah pada keyakinan yang lebih besar. Hal ini menunjukkan bahwa fitur yang dianalisis memberikan bobot yang lebih kuat kepada keputusan untuk membeli laptop.
- ◆ Normalisasi penting dalam konteks ini karena membantu menghasilkan probabilitas yang saling melengkapi, di mana kedua hasil harus menambahkan hingga 1. Hal ini juga menunjukkan bahwa model mampu mengkonversi data awal yang tampaknya tidak mendukung keputusan menjadi hasil yang lebih meyakinkan setelah mempertimbangkan semua informasi.