

MACHINE LEARNING

(Tugas 4)



Disusun Oleh:

Nama: Sifa Maryam Rahman

NPM: 41155050210031

Kelas: TIF A1

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS LANGLANGBUANA

2024

1. Berikut adalah hasil praktik dari video youtube

<https://youtu.be/Sj1ybuDDf9I?si=hCajHe1zasTQ9HGY>

1.1. Pengenalan Bayes Theorem | Teori Bayes | Conditional Probability

Bayes' theorem menawarkan suatu formula untuk menghitung nilai probability dari suatu event dengan memanfaatkan pengetahuan sebelumnya dari kondisi terkait atau sering kali dikenal dengan istilah conditional probability.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$$P(y|X) = \frac{P(X|y) \times P(y)}{P(X)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

Teorema Bayes adalah rumus dalam probabilitas dan statistik yang digunakan untuk menghitung kemungkinan terjadinya suatu peristiwa (event) dengan mempertimbangkan informasi sebelumnya atau kondisi terkait. Ini dikenal juga sebagai conditional probability atau probabilitas bersyarat, di mana kita menghitung kemungkinan dari suatu event (A) terjadi dengan asumsi bahwa event lain (B) sudah terjadi.

- $P(A|B)$: Probabilitas A terjadi jika B diketahui.
- $P(B|A)$: Probabilitas B terjadi jika A diketahui.
- $P(A)$: Probabilitas dari A terjadi secara keseluruhan.
- $P(B)$: Probabilitas dari B terjadi secara keseluruhan.

Dalam aplikasi umum, formula ini sering kali ditulis dalam bentuk posterior = (likelihood \times prior) / evidence, di mana:

- Posterior adalah probabilitas akhir setelah mempertimbangkan bukti baru.
- Likelihood adalah seberapa besar kemungkinan bukti muncul jika hipotesis benar.
- Prior adalah probabilitas awal sebelum adanya bukti baru.
- Evidence adalah total probabilitas bukti di seluruh hipotesis yang ada.

Teorema Bayes digunakan dalam berbagai bidang seperti data mining, machine learning, dan analisis statistik untuk memperbarui pengetahuan tentang suatu event berdasarkan data baru.

1.2. Pengenalan Naive Bayes Classification

Studi Kasus 1

Asep	Joko
+ siomay:0.1	+ siomay: 0.5
+ bakso:0.8	+ bakso: 0.2
+ lumpia: 0.1	+ lumpia: 0.3

Misi: Lakukan prediksi siapa pelanggan yang melakukan pemesanan dengan diketahui pesanannya adalah lumpia dan bakso.

Kasus ini merupakan contoh penerapan Naive Bayes Classification, sebuah metode klasifikasi dalam machine learning yang menggunakan Teorema Bayes dengan asumsi bahwa setiap fitur bersifat independen satu sama lain (dalam hal ini, jenis makanan yang dipesan dianggap tidak saling mempengaruhi).

Di sini, kita ingin memprediksi apakah pelanggan yang memesan lumpia dan bakso adalah Asep atau Joko.

Langkah-langkah Solusi:

1. Hitung Probabilitas Setiap Pelanggan untuk Setiap Pesanan:

➤ Untuk Asep

- Probabilitas memesan lumpia 0.1
- Probabilitas memesan bakso 0.8

➤ Untuk Joko

- Probabilitas memesan lumpia 0.3
- Probabilitas memesan bakso 0.2

2. Kalikan Probabilitas Setiap Fitur

Untuk Asep: $P(\text{Asep}/\text{lumpia}, \text{bakso}) \approx 0.1 \times 0.8 = 0.08$

Untuk Joko: $P(\text{Joko}/\text{lumpia}, \text{bakso}) \approx 0.3 \times 0.2 = 0.06$

3. Kesimpulan

- Karena hasil perhitungan probabilitas Asep lebih tinggi (0.08) dibandingkan dengan Joko (0.06), maka kemungkinan besar pelanggan yang memesan lumpia dan bakso adalah Asep.

1.3. Pengenalan Prior Probability

Prior Probability : $P(y)$

$$P(Asep) = 0.5$$

$$P(Joko) = 0.5$$

1.4. Pengenalan Likelihood

Likelihood : $P(X/y)$

P (Asep)

$$\begin{aligned} P(Lumpia, Bakso/Asep) &= (0.1 \times 0.8) \\ &= 0.08 \end{aligned}$$

P (Joko)

$$\begin{aligned} P(Lumpia, Bakso/Joko) &= (0.3 \times 0.2) \\ &= 0.06 \end{aligned}$$

kita akan melakukan prediksi dengan menggunakan konsep likelihood dalam Naive Bayes Classification untuk menghitung peluang bahwa pesanan "lumpia dan bakso" dibuat oleh Asep atau Joko.

1. Likelihood untuk Asep

- P(Asep) adalah kemungkinan Asep yang memesan.
- P(Lumpia, Bakso | Asep) adalah probabilitas Asep memesan lumpia dan bakso bersama-sama, yang bisa dihitung dengan mengalikan peluang memesan lumpia dan bakso secara terpisah karena asumsi independen.

- Perhitungan untuk Asep

$$P(Lumpia, Bakso/Asep) = P(Lumpia/Asep) \times P(Bakso/Asep)$$

- Substitusi nilai

$$= 0.1 \times 0.8 = 0.08$$

- Jadi, probabilitas (likelihood) bahwa pesanan lumpia dan bakso berasal dari Asep adalah 0.08.

2. Likelihood untuk Joko

- P(Joko) adalah kemungkinan Joko yang memesan.
- P(Lumpia, Bakso | Joko) adalah probabilitas Joko memesan lumpia dan bakso bersama-sama, juga dihitung dengan mengalikan peluang memesan lumpia dan bakso secara terpisah.
- Perhitungan untuk Joko

$$P(Lumpia, Bakso/Joko) = P(Lumpia/Joko) \times P(Bakso/Joko)$$

➤ Substitusi nilai

$$= 0.3 \times 0.2 = 0.06$$

➤ Jadi, probabilitas (likelihood) bahwa pesanan lumpia dan bakso berasal dari Joko adalah 0.06.

Kesimpulan

Berdasarkan nilai likelihood yang dihitung:

- **Asep** memiliki nilai likelihood lebih tinggi (0.08) dibandingkan Joko (0.06).
- Ini menunjukkan bahwa, berdasarkan informasi yang ada, lebih mungkin bahwa pesanan "lumpia dan bakso" berasal dari **Asep**.

1.5. Pengenalan Evidence | Normalizer

Evidence atau Normalizer $P(X)$

$$Evidence = \sum (Likelihood \times Prior)$$

$$\begin{aligned} P(lumpia, bakso) &= (0.08 \times 0.5) + (0.06 \times 0.5) \\ &= 0.07 \end{aligned}$$

Nilai Evidence $P(lumpia, bakso) = 0.07$ mewakili total probabilitas bahwa pesanan lumpia dan bakso terjadi, tanpa mengaitkan siapa yang memesan. Ini adalah nilai normalisasi yang akan kita gunakan untuk menghitung probabilitas posterior setiap pelanggan (Asep atau Joko) setelah mengetahui pesanan.

1.6. Pengenalan Posterior Probability

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Asep

$$\begin{aligned} P(Asep/lumpia, bakso) &= \frac{0.08 \times 0.5}{0.07} \\ &= 0.57 \end{aligned}$$

Joko

$$\begin{aligned} P(Joko/lumpia, bakso) &= \frac{0.06 \times 0.5}{0.07} \\ &= 0.43 \end{aligned}$$

Kita sekarang akan menghitung probabilitas posterior atau $P(\text{Asep} \mid \text{lumpia, bakso})$ dan $P(\text{Joko} \mid \text{lumpia, bakso})$ menggunakan Bayes' Theorem. Probabilitas posterior ini menunjukkan seberapa besar kemungkinan bahwa pelanggan yang memesan lumpia dan bakso adalah Asep atau Joko setelah mempertimbangkan data yang ada (yaitu pesanan lumpia dan bakso).

Formula Posterior

Secara umum, probabilitas posterior dapat dihitung menggunakan formula berikut

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

Interpretasi Hasil

- $P(\text{Asep} \mid \text{lumpia, bakso}) = 0.57$ dan $P(\text{Joko} \mid \text{lumpia, bakso}) = 0.43$.
- Dengan kata lain, setelah mempertimbangkan pesanan lumpia dan bakso, ada **kemungkinan 57%** bahwa pesanan ini berasal dari **Asep** dan **43%** kemungkinan dari **Joko**.
- Karena $57\% > 43\%$, kita dapat menyimpulkan bahwa **Asep** lebih mungkin sebagai pelanggan yang memesan lumpia dan bakso dibandingkan Joko.

1.7. Studi kasus dan implementasi Naive Bayes

```
[1]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.datasets import load_breast_cancer
      print(load_breast_cancer().DESCR)
```

```
sifamaryamrahman-4115505050210031

.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
-----

**Data Set Characteristics:**

:Number of Instances: 569

:Number of Attributes: 30 numeric, predictive attributes and the class

:Attribute Information:
  - radius (mean of distances from center to points on the perimeter)
```

```
[3]: X, y = load_breast_cancer(return_X_y=True)
      X.shape
```

```
[3]: (569, 30)
```

```
[5]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

      print(X_train.shape)
      print(X_test.shape)
```

sifamaryamrahman-4115505050210031

(455, 30)
(114, 30)

```
[6]: print('\n')
      print('sifamaryamrahman-4115505050210031')
      print('\n')

      from sklearn.naive_bayes import GaussianNB
      from sklearn.metrics import accuracy_score

      model = GaussianNB()
      model.fit(X_train, y_train)
      y_pred = model.predict(X_test)
      accuracy_score(y_test, y_pred)
```

sifamaryamrahman-4115505050210031

[6]: 0.9298245614035088

[10]: model.score(X_test, y_test)

[10]: 0.9298245614035088

[]:

