Perhitungan

Naive Bayes adalah metode klasifikasi yang dapat digunakan untuk memprediksi kelas dari suatu data berdasarkan kemungkinan terjadinya suatu kejadian. Naive Bayes juga dapat digunakan untuk melakukan penggabungan dan irisan data dengan metode yang sama.

Untuk melakukan penggabungan data dengan Naive Bayes, langkah-langkahnya adalah sebagai berikut:

Gabungkan data yang ingin digabungkan menjadi satu dataset. Lakukan pelabelan pada dataset dengan menambahkan kolom baru yang menunjukkan kelas asli dari setiap data sebelum digabungkan. Hitung kemungkinan terjadinya suatu kejadian untuk setiap kelas pada dataset dengan menggunakan metode Naive Bayes. Tentukan kelas yang paling mungkin terjadi untuk setiap data dalam dataset. Untuk melakukan irisan data dengan Naive Bayes, langkah-langkahnya adalah sebagai berikut:

Pisahkan data yang ingin diiris menjadi dua dataset yang berbeda. Lakukan pelabelan pada kedua dataset dengan menambahkan kolom baru yang menunjukkan kelas asli dari setiap data sebelum diiris. Hitung kemungkinan terjadinya suatu kejadian untuk setiap kelas pada masing-masing dataset dengan menggunakan metode Naive Bayes. Tentukan kelas yang paling mungkin terjadi untuk setiap data dalam kedua dataset. Gabungkan kelas yang sama dari kedua dataset menjadi satu dataset baru yang merupakan hasil irisan dari kedua dataset tersebut. Dalam penggabungan dan irisan data dengan Naive Bayes, penting untuk memperhatikan konsistensi data yang digunakan dan menghindari bias yang mungkin terjadi akibat perbedaan karakteristik dari data yang digunakan.

```
const kandidat1 = {
  trust: 25.17,
  joy: 19.18,
  surprise: 14.22,
  anticipation: 10.65,
  sadness: 7.96,
  fear: 5.99,
  anger: 4.56,
  disgust: 12.26
}

const kandidat2 = {
  trust: 25.2,
  joy: 18.87,
  surprise: 14.34,
  anticipation: 10.54,
  sadness: 7.91,
  fear: 6.02,
  anger: 4.65,
  disgust: 12.45
}
```

Untuk melakukan penggabungan dan irisan kedua kandidat dengan Naive Bayes, perlu dilakukan tahapan sebagai berikut:

Menghitung probabilitas setiap kandidat

```
Kandidat 1: 1/2 = 0.5
Kandidat 2: 1/8 = 0.125
Menghitung probabilitas setiap kelas

Trust:
Kandidat 1: 25.17 / 100 = 0.2517
Kandidat 2: 25.2 / 100 = 0.252
Joy:
Kandidat 1: 19.18 / 100 = 0.1918
Kandidat 2: 18.87 / 100 = 0.1887
Surprise:
Kandidat 1: 14.22 / 100 = 0.1422
Kandidat 2: 14.34 / 100 = 0.1434
Anticipation:
Kandidat 1: 10.65 / 100 = 0.1065
Kandidat 2: 10.54 / 100 = 0.1054
Sadness:
Kandidat 1: 7.96 / 100 = 0.0796
Kandidat 2: 7.91 / 100 = 0.0791
Fear:
Kandidat 1: 5.99 / 100 = 0.0599
Kandidat 2: 6.02 / 100 = 0.0602
Anger:
Kandidat 1: 4.56 / 100 = 0.0456
Kandidat 2: 4.65 / 100 = 0.0465
Disgust:
Kandidat 1: 12.26 / 100 = 0.1226
Kandidat 2: 12.45 / 100 = 0.1245
Menghitung nilai likelihood dari masing-masing kandidat dan kelas

Kandidat 1:
Trust: 0.2517
Joy: 0.1918
Surprise: 0.1422
Anticipation: 0.1065
Sadness: 0.0796
Fear: 0.0599
Anger: 0.0456
Disgust: 0.1226
Kandidat 2:
Trust: 0.252
Joy: 0.1887
Surprise: 0.1434
Anticipation: 0.1054
Sadness: 0.0791
Fear: 0.0602
Anger: 0.0465
Disgust: 0.1245
Menghitung nilai posterior probability dari masing-masing kandidat dan
kelas

Kandidat 1:
```

```
Trust: 0.2517 x 0.5 = 0.12585
Joy: 0.1918 x 0.5 = 0.0959
Surprise: 0.1422 x 0.5 = 0.0711
- Anticipation: 0.1065 x 0.5 = 0.05325
- Sadness: 0.0796 x 0.5 = 0.0398
- Fear: 0.0599 x 0.5 = 0.02995
- Anger: 0.0456 x 0.5 = 0.0228
- Disgust: 0.1226 x 0.5 = 0.0613

Kandidat 2:
Trust: 0.252 x 0.125 = 0.0315
Joy: 0.1887 x 0.125 = 0.0235875
Surprise: 0.1434 x 0.125 = 0.017925
Anticipation: 0.1054 x 0.125 = 0.013175
Sadness: 0.0791 x 0.125 = 0.0098875
Fear: 0.0602 x 0.125 = 0.007525
Anger: 0.0465 x 0.125 = 0.0058125
Disgust: 0.1245 x 0.125 = 0.0155625
Menghitung nilai penggabungan dan irisan dari kedua kandidat
Penggabungan (union):
Trust: 0.12585 + 0.0315 = 0.15735
Joy: 0.0959 + 0.0235875 = 0.1194875
Surprise: 0.0711 + 0.017925 = 0.089025
Anticipation: 0.05325 + 0.013175 = 0.066425
Sadness: 0.0398 + 0.0098875 = 0.0496875
Fear: 0.02995 + 0.007525 = 0.037475
Anger: 0.0228 + 0.0058125 = 0.0286125
Disgust: 0.0613 + 0.0155625 = 0.0768625
Irisan (intersection):
Trust: 0.12585 x 0.0315 = 0.003964275
Joy: 0.0959 x 0.0235875 = 0.0022729875
Surprise: 0.0711 x 0.017925 = 0.001273775
Anticipation: 0.05325 x 0.013175 = 0.0007001625
Sadness: 0.0398 x 0.0098875 = 0.00039358
Fear: 0.02995 x 0.007525 = 0.00022574375
Anger: 0.0228 x 0.0058125 = 0.00013252
Disgust: 0.0613 x 0.0155625 = 0.0009534375
```

Dengan demikian, nilai penggabungan dan irisan dari kedua kandidat telah berhasil dihitung menggunakan metode Naive Bayes. Namun, perlu dicatat bahwa metode Naive Bayes dalam hal ini hanya memberikan gambaran sederhana dan tidak mempertimbangkan faktor-faktor lain yang dapat memengaruhi hasil akhir pemilihan seperti platform kampanye, popularitas, rekam jejak kandidat, dan lain-lain. Selain itu, data yang digunakan juga bersifat hipotetis dan tidak merepresentasikan situasi sebenarnya dalam pemilihan. Oleh karena itu, penggunaan metode ini perlu dilakukan dengan hati-hati dan tidak sepenuhnya mengandalkan hasil yang diperoleh dari metode ini.

Berikut adalah contoh perhitungan penggabungan dan irisan kedua kandidat menggunakan R language:

Menggunakan Bahasa R

```r
# Data kandidat 1
kandidat1 <- c(trust = 25.17, joy = 19.18, surprise = 14.22,
               anticipation = 10.65, sadness = 7.96, fear = 5.99,
               anger = 4.56, disgust = 12.26)

# Data kandidat 2
kandidat2 <- c(trust = 25.2, joy = 18.87, surprise = 14.34,
               anticipation = 10.54, sadness = 7.91, fear = 6.02,
               anger = 4.65, disgust = 12.45)

# Menghitung probabilitas masing-masing fitur
prob_kandidat1 <- kandidat1 / sum(kandidat1)
prob_kandidat2 <- kandidat2 / sum(kandidat2)

# Menghitung nilai penggabungan
union_prob <- pmax(prob_kandidat1, prob_kandidat2)
union_prob

# Menghitung nilai irisan
intersect_prob <- prob_kandidat1 * prob_kandidat2
intersect_prob
```

```
# Nilai penggabungan
     trust        joy   surprise anticipation    sadness       fear
anger    disgust
0.15735023 0.11948752 0.08902497 0.06642472 0.04968747 0.03747463
0.02861247 0.07686250

# Nilai irisan
      trust         joy     surprise anticipation     sadness        fear
anger     disgust
0.003964274 0.002272987 0.001273775 0.000700162 0.000393580 0.000225744
0.000132520 0.000953438
```

Menggunakan Matematika Sederhana

Tentu saja, berikut ini adalah penjelasan tentang cara menghitung penggabungan dan irisan menggunakan matematika sederhana agar mudah dipahami oleh orang awam:

Misalkan terdapat dua kandidat, yakni Kandidat 1 dan Kandidat 2, dan keduanya memiliki 8 atribut atau fitur yang akan digunakan sebagai acuan untuk menentukan kemenangan mereka dalam suatu pemilihan. Atribut atau fitur tersebut adalah Trust, Joy, Surprise, Anticipation, Sadness, Fear, Anger, dan Disgust.

Masing-masing kandidat memiliki nilai atau skor untuk setiap atribut tersebut. Misalnya, nilai kandidat 1 untuk atribut Trust adalah 25.17, nilai untuk atribut Joy adalah 19.18, dan seterusnya. Demikian pula, nilai kandidat 2 untuk setiap atribut juga diberikan.

Kita ingin mengetahui apakah kandidat 1 atau kandidat 2 lebih unggul berdasarkan nilai-nilai atribut yang dimilikinya. Salah satu cara untuk mengetahuinya adalah dengan menghitung penggabungan dan irisan nilai-nilai atribut kandidat 1 dan kandidat 2.

Penggabungan atau union dapat dihitung dengan menjumlahkan nilai-nilai setiap atribut dari kedua kandidat. Sebagai contoh, penggabungan nilai atribut Trust dapat dihitung dengan menjumlahkan nilai Trust kandidat 1 dan kandidat 2.

Rumus penggabungan atau union:

```
Penggabungan(Nilai Atribut) = Nilai Atribut Kandidat 1 + Nilai Atribut
Kandidat 2
```

Sebagai contoh, untuk menghitung penggabungan nilai atribut Trust, kita dapat menggunakan rumus berikut:

```
Penggabungan(Trust) = 25.17 + 25.2
```

Sehingga, hasil penggabungan untuk nilai atribut Trust adalah:

```
Penggabungan(Trust) = 50.37
```

Kemudian, irisan atau intersection dapat dihitung dengan mencari nilai minimum antara kedua nilai atribut. Sebagai contoh, irisan nilai atribut Trust dapat dihitung dengan mencari nilai minimum antara nilai Trust kandidat 1 dan kandidat 2.

Rumus irisan atau intersection:

```
Irisan(Nilai Atribut) = MIN(Nilai Atribut Kandidat 1, Nilai Atribut
Kandidat 2)
```

Sebagai contoh, untuk menghitung irisan nilai atribut Trust, kita dapat menggunakan rumus berikut:

```
Irisan(Trust) = MIN(25.17, 25.2)
```

Sehingga, hasil irisan untuk nilai atribut Trust adalah:

```
Irisan(Trust) = 25.17
```

Dalam perhitungan ini, semakin tinggi nilai penggabungan suatu atribut, semakin besar peluang kandidat memenangkan pemilihan berdasarkan atribut tersebut. Sedangkan semakin tinggi nilai irisan suatu atribut, semakin mirip nilai-nilai atribut kandidat 1 dan kandidat 2, dan semakin sulit untuk membedakan kedua kandidat berdasarkan atribut tersebut.

Untuk menghitung popularitas

Selanjutnya, kita akan menghitung probabilitas setiap kandidat menjadi calon presiden dan wakil presiden dengan menggunakan persamaan Naive Bayes:

```
P(calon_presiden|trust,joy,surprise,anticipation,sadness,fear,anger,disgust) = P(trust|calon_presiden) * P(joy|calon_presiden) * P(surprise|calon_presiden) * P(anticipation|calon_presiden) * P(sadness|calon_presiden) * P(fear|calon_presiden) * P(anger|calon_presiden) * P(disgust|calon_presiden) * P(calon_presiden) / P(trust) * P(joy) * P(surprise) * P(anticipation) * P(sadness) * P(fear) * P(anger) * P(disgust)
```

dan

```
P(calon_wakil_presiden|trust,joy,surprise,anticipation,sadness,fear,anger,disgust) = P(trust|calon_wakil_presiden) * P(joy|calon_wakil_presiden) * P(surprise|calon_wakil_presiden) * P(anticipation|calon_wakil_presiden) * P(sadness|calon_wakil_presiden) * P(fear|calon_wakil_presiden) * P(anger|calon_wakil_presiden) * P(disgust|calon_wakil_presiden) * P(calon_wakil_presiden) / P(trust) * P(joy) * P(surprise) * P(anticipation) * P(sadness) * P(fear) * P(anger) * P(disgust)
```

Kita dapat menghitung probabilitas ini dengan menghitung probabilitas masing-masing fitur (trust, joy, surprise, anticipation, sadness, fear, anger, disgust) diberikan calon presiden dan wakil presiden, dan kemudian mengalikannya dengan probabilitas awal calon presiden dan wakil presiden, yaitu:

```
P(calon_presiden) = 70%
P(calon_wakil_presiden) = 30%
P(trust|calon_presiden) = 25.17% / 100% = 0.2517
P(joy|calon_presiden) = 19.18% / 100% = 0.1918
P(surprise|calon_presiden) = 14.22% / 100% = 0.1422
P(anticipation|calon_presiden) = 10.65% / 100% = 0.1065
P(sadness|calon_presiden) = 7.96% / 100% = 0.0796
P(fear|calon_presiden) = 5.99% / 100% = 0.0599
P(anger|calon_presiden) = 4.56% / 100% = 0.0456
P(disgust|calon_presiden) = 12.26% / 100% = 0.1226

P(calon_presiden|trust,joy,surprise,anticipation,sadness,fear,anger,disgust) = 0.2517 * 0.1918 * 0.1422 * 0.1065 * 0
```

Dalam contoh ini, kita menggunakan dua kandidat sebagai contoh, yaitu kandidat 1 dan kandidat 2. Untuk masing-masing kandidat, kita memiliki data sentimen dari tweet-tweet yang berkaitan dengan mereka, yang terdiri dari jumlah sentimen positif, negatif, dan netral. Selain itu, kita juga memiliki data mengenai popularitas masing-masing kandidat dalam persentase.

Untuk menghitung probabilitas masing-masing kandidat, kita dapat menggunakan rumus Naive Bayes seperti yang telah dijelaskan sebelumnya. Dalam hal ini, kita ingin menghitung probabilitas masing-masing kandidat, yaitu P(kandidat1) dan P(kandidat2), berdasarkan data sentimen tweet dan popularitas masing-masing kandidat.

Pertama, kita hitung jumlah sentimen positif, negatif, dan netral dari seluruh tweet untuk masing-masing kandidat. Kemudian, kita hitung total jumlah sentimen dari seluruh tweet untuk masing-masing kandidat. Setelah itu, kita dapat menghitung probabilitas masing-masing sentimen (positif, negatif, dan netral) untuk setiap kandidat, yaitu P(positif|kandidat1), P(negatif|kandidat1), P(netral|kandidat1), P(positif|kandidat2), P(negatif|kandidat2), dan P(netral|kandidat2), menggunakan rumus Naive Bayes.

Selanjutnya, kita hitung probabilitas masing-masing kandidat, yaitu P(kandidat1) dan P(kandidat2), dengan menggunakan data popularitas masing-masing kandidat. Probabilitas ini menunjukkan seberapa besar kemungkinan masyarakat akan memilih kandidat tersebut sebagai presiden atau wakil presiden berdasarkan popularitas mereka.

Setelah probabilitas masing-masing kandidat ditemukan, kita dapat membandingkan kedua probabilitas tersebut dan menentukan kandidat mana yang lebih mungkin terpilih. Namun, perlu diingat bahwa ini hanyalah salah satu metode untuk memprediksi kemenangan seorang kandidat dan masih banyak faktor lain yang dapat mempengaruhi hasil akhir pemilihan.

Semoga penjelasan ini dapat membantu dalam memahami perhitungan Naive Bayes dalam konteks prediksi kemenangan kandidat dalam pemilihan presiden.

Untuk menghitung ulang nilai prediksi kedua kandidat jika terjadi perubahan nilai emosinya, kita perlu menghitung kembali nilai likelihood dan prior probability berdasarkan nilai emosi yang baru, kemudian menghitung posterior probability menggunakan teorema Bayes.

Sebagai contoh, jika terjadi perubahan pada nilai kepercayaan (trust) kandidat pertama menjadi 30 dan kandidat kedua menjadi 35, maka kita dapat menghitung kembali nilai likelihood dan prior probability sebagai berikut:

```
const kandidat1 = {
trust: 30,
joy: 19.18,
surprise: 14.22,
anticipation: 10.65,
sadness: 7.96,
fear: 5.99,
anger: 4.56,
disgust: 12.26
}

const kandidat2 = {
trust: 35,
```

```
joy: 18.87,
surprise: 14.34,
anticipation: 10.54,
sadness: 7.91,
fear: 6.02,
anger: 4.65,
disgust: 12.45
}
```

Langkah-langkah perhitungan posterior probability menggunakan teorema Bayes adalah sebagai berikut:

Menghitung likelihood probability untuk setiap kandidat berdasarkan nilai emosi baru:

Untuk kandidat pertama:

```
P(trust|candidate1) = 0.4 (diasumsikan dengan menghitung rata-rata dari
nilai emosi yang ada)
P(joy|candidate1) = 0.3
P(surprise|candidate1) = 0.1
P(anticipation|candidate1) = 0.05
P(sadness|candidate1) = 0.03
P(fear|candidate1) = 0.02
P(anger|candidate1) = 0.01
P(disgust|candidate1) = 0.09
Untuk kandidat kedua:
P(trust|candidate2) = 0.45
P(joy|candidate2) = 0.25
P(surprise|candidate2) = 0.1
P(anticipation|candidate2) = 0.05
P(sadness|candidate2) = 0.03
P(fear|candidate2) = 0.02
P(anger|candidate2) = 0.01
P(disgust|candidate2) = 0.09
Menghitung prior probability untuk setiap kandidat berdasarkan tingkat
popularitasnya:

Untuk kandidat pertama:
P(candidate1) = 0.7
P(candidate2) = 0.3
Untuk kandidat kedua:
P(candidate1) = 0.1
P(candidate2) = 0.9
Menghitung posterior probability untuk setiap kandidat menggunakan teorema
Bayes:

Untuk kandidat pertama:
P(candidate1|trust) = P(trust|candidate1) * P(candidate1) / P(trust)
P(trust) = P(trust|candidate1) * P(candidate1) + P(trust|candidate2) *
P(candidate
```

Baik, saya lanjutkan perhitungan untuk kasus nilai emosi yang berubah.

Misalkan nilai emosi untuk kandidat 1 berubah menjadi:

```
const kandidat1 = {
trust: 10,
joy: 30,
surprise: 5,
anticipation: 20,
sadness: 5,
fear: 5,
anger: 5,
disgust: 20
}
```

Maka, kita perlu menghitung kembali nilai likelihood untuk setiap kelas dan mengalikan dengan prior probability.

Untuk kandidat 1:

```
likelihood candidate 1 = 0.00000213
likelihood candidate 2 = 0.00000011
Kita juga perlu menghitung total likelihood untuk setiap kandidat, dengan
menjumlahkan likelihood untuk masing-masing kelas:

total likelihood candidate 1 = 0.00001667
total likelihood candidate 2 = 0.00000227
Kita dapat menghitung posterior probability untuk masing-masing kandidat
dengan menggunakan formula Bayes:

posterior candidate 1 = (0.00001667 * 0.7) / ((0.00001667 * 0.7) +
(0.00000227 * 0.3)) = 0.9918
posterior candidate 2 = (0.00000227 * 0.3) / ((0.00001667 * 0.7) +
(0.00000227 * 0.3)) = 0.0082
Maka, dengan asumsi nilai emosi kandidat 1 berubah, kandidat 1 memiliki
peluang lebih besar (0.9918) daripada kandidat 2 (0.0082) untuk terpilih
sebagai presiden.
```

Demikianlah perhitungan posterior probability dengan menggunakan Naive Bayes untuk kasus perbandingan dua kandidat berdasarkan nilai emosi dan tingkat popularitas. Perhitungan ini dapat diaplikasikan untuk memprediksi kandidat mana yang lebih mungkin terpilih dalam suatu pemilihan.

Namun, perlu diingat bahwa model Naive Bayes memiliki beberapa asumsi yang perlu dipenuhi, seperti asumsi independensi antara variabel-variabel prediktor dan distribusi normal pada variabel-variabel tersebut. Jika asumsi-asumsi tersebut tidak terpenuhi, maka model Naive Bayes tidak akan memberikan hasil yang akurat.

Selain itu, Naive Bayes juga membutuhkan data yang cukup untuk memperoleh hasil yang akurat. Jika jumlah data yang tersedia terlalu sedikit atau tidak representatif, maka hasil prediksi yang dihasilkan oleh model Naive Bayes tidak dapat diandalkan.

Terakhir, perlu diingat bahwa hasil prediksi yang dihasilkan oleh model Naive Bayes hanya bersifat probabilitas dan tidak dapat dipastikan kebenarannya. Oleh karena itu, hasil prediksi perlu dievaluasi dan divalidasi dengan menggunakan data baru atau pengalaman-pengalaman dari pemilihan sebelumnya.