**PENGENALAN METODE KLASIFIKASI**

**DECISSION TREE (ID3, C4.5, C5.0)**



**Aryajaya Alamsyah, S.Kom**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA**

**TEKNIK INFORMATIKA**

**2020/2021**

Klasifikasi adalah suatu teknik dalam data mining yang bertujuan untuk mengelompokan data kedalam kelas-kelas yang telah memiliki label. Terdapat beberapa metode klasifikasi yang dapat digunakan, sebagai contoh Decision Tree, Naïve Bayes, Support Vector Machine (SVM), Random Forest, Neural Network, K-Nearest Neighbor (K-NN) dan lain-lain. Namun pembahasan kali ini akan membahas metode Decision Tree atau dibeberapa literatur disebut sebagai ID3.

ID3 mengalami perkembangan menjadi C45 dan C50. ID3 pertama kali diusulkan oleh Quinlan tahun 1986. C45 pertama kali oleh Quinlan tahun 1993. C50 pertama kali diusulkan oleh Quinlan, Cubist, Cestnik tahun 1995. ID3 cocok digunakan untuk dataset yang bertipe nominal dan ordinal. C45 dan C50 cocok digunakan untuk dataset yang bertipe nominal, ordinal dan numerik. Hasil klasifikasi ID3, C45, C50 berupa pohon keputusan. Dari hasil pohon keputusan dapat membentuk aturan fuzzy. Ketiga metode tersebut bekerja dengan melakukan evaluasi semua variabel menggunakan suatu ukuran ketidak murnian, seperti Information Gain dan Gain Ratio untuk mengukur efektifitas suatu variabel dalam mengklasifikasikan himpunan sampel data. Metode ID3 menggunakan Informasi Gain. C45 menggunakan Gain Ratio. C50 menggunakan Gini Index, Adaboost, Boosting.

Entropy adalah suatu parameter untuk mengukur heterogenitas dalam suatu himpunan data. Rumus entropy adalah:

S = Himpunan kasus

N = jumlah nilai yang terdapat pada variabel target

= rasio antara jumlah sampel dikelas *i* dengan jumlah semua sampel pada himpunan data.

Information Gain adalah ukuran efektivitas suatu variabel dalam mengklasifikasikan data.

S = sample (data) space

A = variables used

V = a possible value for variable A

Value(A)= the set of possible values for variable A

= the number of samples for the value v

= the number of all data samples

Gain Ratio.

Penggunaan Entropy sebagai metrik dalam membangun pohon keputusan memiliki masalah bahwa Information Gain ternyata bias terhadap variabel yang memiliki banyak nilai. Semakin banyak nilai dari variabel, maka semakin besar pula nilai Entropy. Untuk menyelesaikan masalah bias, maka digunakan konsep Gain Ratio yang merupakan konsep normalisasi terhadap nilai Information Gain. Gain Ratio dibentuk dari Split Information.

Studi Kasus Klasifikasi Decision Tree.

klasifikasi variabel untuk menentukan apakah dapat berpergian atau tidak. Outlook, temperature, humidity, wind adalah variabel bebas. Travel adalah variabel terikat (label).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Outlook | Temperature | Humidity | Wind | Travel |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Overcast | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Rainy | Cool | Normal | False | Yes |
| 6 | Rainy | Cool | Normal | True | Yes |
| 7 | Overcast | Cool | Normal | True | Yes |
| 8 | Sunny | Mild | High | False | No |
| 9 | Sunny | Cool | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | False | Yes |
| 11 | Sunny | Mild | Normal | True | Yes |
| 12 | Overcast | Mild | High | True | Yes |
| 13 | Overcast | Hot | Normal | False | Yes |
| 14 | Rainy | Mild | High | True | No |

Langkah-Langkah Algoritma Decision Tree

1. Hitung Entropy Total dan Entropy seluruh variabel
2. Hitung Information Gain untuk setiap variabel
3. Variabel dengan Information Gain tertinggi akan menjadi split node
4. Filter dataset menurut split node dan ulangi langkah 2 dan 3 hingga seluruh variabel terpilih atau seluruh data habis dibagi.

Langkah 1. Hitung nilai Entropy Total dan Entropy seluruh variabel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Jumlah Data | Yes | No | Entropy | Information Gain |
| Total |  | 14 | 10 | 4 | 0.863 |  |
| Outlook |  |  |  |  |  |  |
|  | Cloudy | 4 | 4 | 0 | 0 |  |
|  | Rainy | 5 | 4 | 1 | 0.722 |  |
|  | Sunny | 5 | 2 | 3 | 0.970 |  |
| Temperature |  |  |  |  |  |  |
|  | Cool | 4 | 0 | 4 | 0 |  |
|  | Hot | 4 | 2 | 2 | 1 |  |
|  | Mild | 6 | 2 | 4 | 0.918 |  |
| Humidity |  |  |  |  |  |  |
|  | High | 7 | 4 | 3 | 0.985 |  |
|  | Normal | 7 | 7 | 0 | 0 |  |
| Wind |  |  |  |  |  |  |
|  | False | 8 | 2 | 6 | 0.811 |  |
|  | True | 6 | 4 | 2 | 0.918 |  |

Entropy Total

Entropy Outlook (Cloudy, Rainy, Sunny)

Entropy Temperature (Cool, Hot, Mild)

Entropy Humidity (High, Normal)

Entropy Wind (False, True)

Langkah 2 Hitung nilai Information Gain untuk setiap variabel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Jumlah Data | Yes | No | Entropy | Information Gain |
| Total |  | 14 | 10 | 4 | 0.863 |  |
| Outlook |  |  |  |  |  | 0.259 |
|  | Cloudy | 4 | 4 | 0 | 0 |  |
|  | Rainy | 5 | 4 | 1 | 0.722 |  |
|  | Sunny | 5 | 2 | 3 | 0.970 |  |
| Temperature |  |  |  |  |  | 0.184 |
|  | Cool | 4 | 0 | 4 | 0 |  |
|  | Hot | 4 | 2 | 2 | 1 |  |
|  | Mild | 6 | 2 | 4 | 0.918 |  |
| Humidity |  |  |  |  |  | 0.370 |
|  | High | 7 | 4 | 3 | 0.985 |  |
|  | Normal | 7 | 7 | 0 | 0 |  |
| Wind |  |  |  |  |  | 0.006 |
|  | False | 8 | 2 | 6 | 0.811 |  |
|  | True | 6 | 4 | 2 | 0.918 |  |

Information gain tertinggi adalah humidity sebesar 0.370. Dengan demikian humidiy dapat menjadi root node. Dari kedua nilai variabel tersebut, nilai variabel humidity (normal) sudah mengklasifikasikan data menjadi keputusan yaitu Travel (Yes), sehingga tidak perlu dilakukan perhitungan lebih lanjut lagi. Namun untuk variabel humidity (high) masih perlu dilakukan perhitungan lebih lanjut.



Gambar 1. Decision Tree Iterasi ke-1

Setelah terbentuk humidity sebagai root pohon, kita filter dataset, ambil baris yang memiliki nilai humidity sama dengan high saja. Dari 14 baris yang ada, menjadi sisa 7 baris.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Outlook | Temperature | Humidity | Wind | Travel |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Cloudy | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Sunny | Mild | High | False | No |
| 6 | Cloudy | Mild | High | True | Yes |
| 7 | Rainy | Mild | High | True | No |

Langkah 3. Hitung kembali nilai Entropy dan Information Gain dari sisa baris yang ada.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Jumlah Data | Yes | No | Entropy | Information Gain |
| Humidity (High) | | 7 | 3 | 4 | 0.98523 |  |
| Outlook |  |  |  |  |  | 0.69951 |
|  | Cloudy | 2 | 2 | 0 | 0 |  |
|  | Rainy | 2 | 1 | 1 | 1 |  |
|  | Sunny | 3 | 0 | 3 | 0 |  |
| Temperature |  |  |  |  |  | 0.02024 |
|  | Cool | 0 | 0 | 0 | 0 |  |
|  | Hot | 3 | 1 | 2 | 0.91830 |  |
|  | Mild | 4 | 2 | 2 | 1 |  |
| Wind |  |  |  |  |  | 0.02024 |
|  | False | 4 | 2 | 2 | 1 |  |
|  | True | 3 | 1 | 2 | 0.91830 |  |

Berdasarkan tabel diatas terlihat bahwa variabel berikutnya yang memiliki information gain tertinggi adalah variabel outlook. Maka variabel outlook menjadi node berikutnya dari pohon yang kita buat. Untuk nilai outlook (Cloudy) sudah mengklasifikasikan data menjadi keputusan Travel (Yes) dan untuk nilai outlook (sunny) juga sudah mengklasifikasikan data menjadi keputusan Travel (No). Namun outlook (rainy) belum dapat diklasifikasikan sehingga perlu dilakukan perhitungan lebih lanjut.



Gambar 2. Decision Tree Iterasi ke-2

Setelah outlook menjadi node berikutnya. Maka kita filter dataset dan ambil baris yang memiliki nilai outlook sama dengan cloudy dan outlook sama dengan Sunny. Dari 5 baris yang ada, menjadi sisa 2 baris.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Outlook | Temperature | Humidity | Wind | Travel |
| 1 | Rainy | Mild | High | False | Yes |
| 2 | Rainy | Mild | High | True | No |

Langkah 4. Hitung kembali nilai Entropy dan Information Gain dari sisa baris yang ada.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Jumlah Data | Yes | No | Entropy | Information Gain |
| Humidity (High) And Outlook (Rainy) | | 2 | 1 | 1 | 1 |  |
| Temperature |  |  |  |  |  | 0 |
|  | Cool | 0 | 0 | 0 | 0 |  |
|  | Hot | 0 | 0 | 0 | 0 |  |
|  | Mild | 2 | 1 | 1 | 1 |  |
| Wind |  |  |  |  |  | 1 |
|  | False | 1 | 1 | 0 | 0 |  |
|  | True | 1 | 0 | 1 | 0 |  |

Berdasarkan tabel diatas, terlihat bahwa variabel berikutnya yang memiliki information gain tertinggi adalah variabel wind. Maka variabel wind menjadi node berikutnya dari pohon yang kita buat. Untuk nilai wind (false) sudah mengklasifikasikan data menjadi keputusan Travel (Yes) dan untuk nilai wind (true) juga sudah mengklasifikasikan data menjadi keputusan Travel (No). Karena semua data sudah habis diklasifikasikan, maka pohon akhir menjadi seperti gambar 3.



Gambar 3. Hasil Decission Tree

Berdasarkan hasil decision tree dapat diketahui beberapa aturan fuzzy untuk menentukan apakah dapat berpergian atau tidak. Sebagai contoh

1. IF Humidity = Normal THEN Travel = Yes
2. IF Humidity = High AND Outlook = Cloudy THEN Travel = Yes
3. IF Humidity = High AND Outlook = Sunny THEN Travel = No
4. IF Humidity = High AND Outlook = Rainy AND Wind = No THEN Travel = Yes
5. IF Humidity = High AND Outlook = Rainy AND Wind = Yes THEN Travel = No

Implementasi Bahasa R.

|  |
| --- |
| library**(**readxl**)**  library**(**party**)**  library**(**rpart**)**  library**(**C50**)**  library**(**Rweka**)**  library**(**partykit**)**  dataset**<-** read\_excel**(**"D:/# Mengajar/Bu Nur/2020-2021 Big Data/Materi/  Pertemuan Klasifikasi/Dataset\_klasifikasi.xlsx"**)**  dataset **<-** data.frame**(**  Outlook **=** as.factor**(**dataset**$**Outlook**)**,  Temperature **=** as.factor**(**dataset**$**Temperature**)**,  Humadity **=** as.factor**(**dataset**$**Humidity**)**,  Wind **=** as.factor**(**dataset**$**Wind**)**,  Travel **=** as.factor**(**dataset**$**Travel**)**  **)**  dataset  formula\_c45 **<-** dataset**$**Travel **~** dataset**$**Outlook **+** dataset**$**Temperature  **+** dataset**$**Humadity **+** dataset**$**Wind  C45 **<-** J48**(**formula\_c45, data**=**dataset**)**  summary**(**C45**)**  plot**(**C45,type**=**"simple"**)**  print**(**oneTree**)** |

Hasil Plot Rstudio

