LAPORAN PRAKTIKUM KECERDASAN BUATAN

ANALISIS PREDIKSI HEART DISEASE MENGGUNAKAN ALGORITMA C4.5

Masayu Franstika[1], Gregorius Gama[2], Junpito Salim [3], Mochammad Aditya Putra Suhendar[4]

Program Studi Sains Data, Jurusan Sains, Institut Teknologi Sumatera

Email : masayu.120450016@student.itera.ac.id[1], gregorius.120450018@student.itera.ac.id[2]

junpito.120450086@student.itera.ac.id[3], mochammad.120450058@student.itera.ac.id[4]

Abstrak

Data mining adalah teknik untuk mengeksplorasi data yang kompleks dan besar untuk menemukan pola yang berguna. Salah satu teknik pengolahan data mining adalah klasifikasi, dan metode klasifikasi yang populer adalah pohon keputusan. Konsep dasar dari pohon keputusan adalah mengubah data menjadi model pohon keputusan, kemudian mengubah model pohon menjadi aturan, dan menyederhanakan aturan, yang dapat direpresentasikan dalam bentuk tabel dengan atribut dan catatan. Ada beberapa cara untuk membangun sebuah pohon keputusan, salah satunya adalah dengan menggunakan algoritma C4.5. Algoritma C4.5 adalah metode berbasis pohon keputusan. Pada algoritma C4.5, pemilihan atribut dilakukan dengan menggunakan gain, ratio, dengan mencari nilai entropi. Topik penelitian kami adalah analisis penyakit jantung. Penyakit jantung merupakan penyebab kematian nomor satu di dunia. Sekitar 17,9 juta orang meninggal karena penyakit jantung pada tahun 2019, terhitung 32% dari kematian populasi dunia.

Kata kunci : Data mining, Pohon keputusan, Algoritma C4.5, Penyakit jantung

1. Pendahuluan

Kesehatan adalah keadaan sehat fisik, mental dan sosial yang utuh, bukan hanya bebas dari penyakit atau kelemahan. Kesehatan merupakan salah satu faktor terpenting yang harus dijaga oleh setiap individu, agar dalam menjalani aktivitas sehari– hari menjadi lebih produktif, tidak mudah lelah. Berbagai faktor sosial mempengaruhi kondisi kesehatan, seperti perilaku pribadi, status sosial, genetika dan biologi, perawatan kesehatan, dan lingkungan fisik. Menjaga kesehatan dan kebugaran tubuh merupakan hal yang sangat penting. Hal ini karena dengan memiliki tubuh yang sehat dan bugar dapat mencegah tubuh terserang penyakit sehingga kita dapat tetap menjalankan aktivitas sehari-hari. Namun gejala penyakit yang tidak terdeteksi dari dini dapat berakibat fatal pada kesehatan. PTM atau Penyakit Tidak Menular menjadi penyebab utama kematian. Setiap tahunnya terdapat lebih dari 36 juta orang yang meninggal disebabkan PTM atau setara 63% dari total seluruh kematian. Penyakit yang rentan terjadi terutama saat seorang individu berada pada usia produktif yaitu penyakit jantung (Heart Disease). Biasanya tidak ada gejala untuk penyakit jantung namun ada gejala untuk serangan jantung dan stroke yaitu rasa sakit atau tidak nyaman di daerah dada dan/atau rasa sakit atau tidak nyaman pada tangan, pundak, rahang, dan punggung. Tingginya faktor kematian dari penyakit jantung karena kurangnya pengetahuan masyarakat terhadap gejala atau tanda–tanda saat seseorang tersebut mengidap penyakit ini.

Industri kesehatan memiliki sejumlah besar data kesehatan, namun sebagian besar data tersebut tidak diolah untuk mengetahui informasi tersembunyi untuk dijadikan pengambilan keputusan yang efektif oleh para praktisi kesehatan. Pengambilan keputusan atas dasar data dan informasi yang akurat akan menghasilkan keputusan dan prediksi penyakit menjadi tepat sasaran. Besarnya jumlah data merupakan sumber daya utama yang akan diproses dan dianalisis guna diekstraksi pengetahuan yang memungkinkan dukungan untuk penghematan biaya dan pengambilan keputusan. Dengan data mining, kita dapat melakukan pengklasifikasian, memprediksi, memperkirakan dan mendapatkan informasi lain yang bermanfaat dari kumpulan data dalam jumlah yang besar, salah satu caranya adalah dengan menggunakan Klasifikasi dalam data mining yang dapat dilakukan dengan menggunakan algoritma C4.5. Dengan algoritma C4.5, akan didapatkan sebuah pohon keputusan yang mudah dipahami dan mudah dimengerti.

Algoritma C4.5 sendiri menggunakan pendekatan induksi dimana dalam pendekatan ini, algoritma C4.5 membagi data berdasarkan kriteria yang dipilih untuk membuat sebuah pohon keputusan yang menggunakan pendekatan secara top-down. Pemilihan algoritma C4.5 adalah, algoritma tersebut mampu menghasilkan subsistem model base yang dapat digunakan untuk menunjang sistem pendukung keputusan. Untuk mendukung pengembangan algoritma C4.5, digunakan metode RGFDT (*Rules Generation From the Decision Tree*) untuk membangun general rules dari rule set yang dihasilkan dari algoritma C4.5.

2. Metode

a. Analisis Masalah

Data yang kita gunakan disini adalah data dari sumber open source. Data ini mempunyai 76 atribut. Namun pada praktikum ini kita hanya menggunakan 14 atribut. Tujuan dari percobaan ini berupa acuan pada penyakit jantung pasien. Nilainya berupa bilangan bulat dari 0 sampai 4. Percobaan yang dilakukan adalah menghasilkan angka lain yang dimulai dari absence value(0)

b. Identifikasi Variabel

Dari Analisis permasalahan diatas dapat diidentifikasi variabel-variabel yang akan terlibat dalam solusi sebagai berikut:

- Variabel himpunan kasus
- Atribut Jumlah partisi atribut
- Jumlah kasus
- Nilai entropy

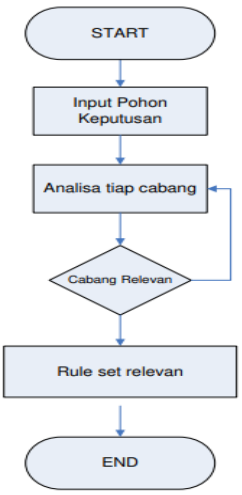
$$\text{Entropy(S)} = \sum_{i=1}^{n} pi * log2pi$$

- Gain

$$\text{Gain(S,A)} = \text{Entropy(S)} - \sum_{i=0}^{n} \frac{|si|}{|s|} Entropy(Si)$$

c. Pengembangan Metode

Flowchart dibawah ini merupakan alur dari metode RGFDT (*Rules Generation From the Decision Tree*). Input berupa pohon keputusan selanjutnya dilakukan analisa tiap cabang, apabila terdapat kondisi yang tidak relevan makan dilakukan pemangkasan cabang. Sehingga menghasilkan rules baru tanpa kondisi yang tidak relevan.



d. Pseudocode

1. import library

```
[1]  import pandas as pd
     import numpy as np
```

2. Load dataset heart.csv dan memasukan kedalam variabel “df”

```
df = pd.read_csv("/content/gdrive/MyDrive/praktikum/heart.csv")
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

3. Deskripsi umum dan Analisis Awal

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

4. Train Data

```
[ ]  df = df.to_numpy()
```

```
[ ]  df
```

```
array([[52.,  1.,  0., ...,  2.,  3.,  0.],
       [53.,  1.,  0., ...,  0.,  3.,  0.],
       [70.,  1.,  0., ...,  0.,  3.,  0.],
       ...,
       [47.,  1.,  0., ...,  1.,  2.,  0.],
       [50.,  0.,  0., ...,  0.,  2.,  1.],
       [54.,  1.,  0., ...,  1.,  3.,  0.]])
```

```
dataTraining = np.concatenate((df[0:40, :], df[50:90, :]),
                              axis=0)
dataTesting = np.concatenate((df[40:50, :], df[90:100, :]),
                               axis=0)
```

Diinputkan dataTraining dan dataTesting dengan ketentuan dataframenya

```
print(dataTraining)
len(dataTraining)
```

```
[[52.  1.  0. ...  2.  3.  0.]
 [53.  1.  0. ...  0.  3.  0.]
 [70.  1.  0. ...  0.  3.  0.]
 ...
 [59.  0.  0. ...  0.  2.  0.]
 [62.  0.  0. ...  2.  2.  0.]
 [68.  1.  0. ...  2.  3.  0.]]
80
```

Output dataTraining

```
print(dataTesting)
len(dataTesting)
```

```
[[6.50e+01 0.00e+00 2.00e+00 1.60e+02 3.60e+02 0.00e+00 0.00e+00 1.51e+02
  0.00e+00 8.00e-01 2.00e+00 0.00e+00 2.00e+00 1.00e+00]
 [5.40e+01 1.00e+00 2.00e+00 1.20e+02 2.58e+02 0.00e+00 0.00e+00 1.47e+02
  0.00e+00 4.00e-01 1.00e+00 0.00e+00 3.00e+00 1.00e+00]
 [6.10e+01 0.00e+00 0.00e+00 1.30e+02 3.30e+02 0.00e+00 0.00e+00 1.69e+02
  0.00e+00 0.00e+00 2.00e+00 0.00e+00 2.00e+00 0.00e+00]
 [4.60e+01 1.00e+00 0.00e+00 1.20e+02 2.49e+02 0.00e+00 0.00e+00 1.44e+02
  0.00e+00 8.00e-01 2.00e+00 0.00e+00 3.00e+00 0.00e+00]
 [5.50e+01 0.00e+00 1.00e+00 1.32e+02 3.42e+02 0.00e+00 1.00e+00 1.66e+02
  0.00e+00 1.20e+00 2.00e+00 0.00e+00 2.00e+00 1.00e+00]
 [4.20e+01 1.00e+00 0.00e+00 1.40e+02 2.26e+02 0.00e+00 1.00e+00 1.78e+02
  0.00e+00 0.00e+00 2.00e+00 0.00e+00 2.00e+00 1.00e+00]
 [4.10e+01 1.00e+00 1.00e+00 1.35e+02 2.03e+02 0.00e+00 1.00e+00 1.32e+02
  0.00e+00 0.00e+00 1.00e+00 0.00e+00 1.00e+00 1.00e+00]
 [6.60e+01 0.00e+00 0.00e+00 1.78e+02 2.28e+02 1.00e+00 1.00e+00 1.65e+02
  1.00e+00 1.00e+00 1.00e+00 2.00e+00 3.00e+00 0.00e+00]
 [6.60e+01 0.00e+00 2.00e+00 1.46e+02 2.78e+02 0.00e+00 0.00e+00 1.52e+02
  0.00e+00 0.00e+00 1.00e+00 1.00e+00 2.00e+00 1.00e+00]
 [6.00e+01 1.00e+00 0.00e+00 1.17e+02 2.30e+02 1.00e+00 1.00e+00 1.60e+02
  1.00e+00 1.40e+00 2.00e+00 2.00e+00 3.00e+00 0.00e+00]
 [5.40e+01 0.00e+00 2.00e+00 1.08e+02 2.67e+02 0.00e+00 0.00e+00 1.67e+02
  0.00e+00 0.00e+00 2.00e+00 0.00e+00 2.00e+00 1.00e+00]
 [6.20e+01 0.00e+00 0.00e+00 1.24e+02 2.09e+02 0.00e+00 1.00e+00 1.63e+02
  0.00e+00 0.00e+00 2.00e+00 0.00e+00 2.00e+00 1.00e+00]
 [6.30e+01 1.00e+00 0.00e+00 1.40e+02 1.87e+02 0.00e+00 0.00e+00 1.44e+02
  1.00e+00 4.00e+00 2.00e+00 2.00e+00 3.00e+00 0.00e+00]
 [4.40e+01 1.00e+00 0.00e+00 1.20e+02 1.69e+02 0.00e+00 1.00e+00 1.44e+02
  1.00e+00 2.80e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00]
 [6.20e+01 1.00e+00 1.00e+00 1.28e+02 2.08e+02 1.00e+00 0.00e+00 1.40e+02
```

Output dataTesting

```
inputTraining = dataTraining[:, 0:4]
inputTesting = dataTesting[:, 0:4]
labelTraining = dataTraining[:, 4]
labelTesting = dataTesting[:, 4]
print(labelTraining)
len(labelTraining)
```

```
[212. 203. 174. 203. 294. 248. 318. 289. 249. 286. 149. 341. 210. 298.
 204. 210. 308. 266. 244. 211. 185. 223. 208. 252. 209. 307. 233. 319.
 256. 327. 169. 244. 131. 269. 196. 231. 213. 271. 263. 229. 283. 241.
 175. 188. 217. 217. 193. 245. 212. 232. 204. 278. 299. 212. 204. 288.
 197. 315. 215. 164. 326. 207. 249. 177. 256. 257. 255. 187. 201. 201.
 233. 149. 231. 175. 215. 220. 211. 249. 268. 193.]
80
```

5. Pemodelan

```
from sklearn import tree
model = tree.DecisionTreeClassifier()
```

Dinputkan Decision Tree Classifier pada model

```
model = model.fit(inputTraining, labelTraining)
```

‘inputTraing’ dan ‘labelTraining’ diinputkan ke model.fit yang dijadikan model

3. Hasil

```
hasilPrediksi = model.predict(inputTesting)
print("Label Sebenarnya : ", labelTesting)
print("Hasil Prediksi : ", hasilPrediksi)
```

```
Label Sebenarnya :  [360. 258. 330. 249. 342. 226. 203. 228. 278. 230. 267. 209. 187. 169.
 208. 236. 303. 282. 248. 197.]
Hasil Prediksi :  [269. 197. 294. 249. 217. 315. 315. 164. 278. 318. 248. 294. 187. 169.
 263. 341. 241. 286. 318. 185.]
```

Dalam ‘hasil prediksi” terdapat model yang telah dibuat berbentuk ‘model.predict’ yang kemudian ditampilkan Label sebenarnya yang diambil dari ‘label testing’ dan Hasil Prediksi yang diambil dari ‘hasil prediksi’. Keduanya menghasilkan output angka beragam dengan retang diatas 150 - 400.

```
prediksiBenar = (hasilPrediksi == labelTesting).sum()
prediksiSalah = (hasilPrediksi != labelTesting).sum()
print("Prediksi Benar : ", prediksiBenar, "data")
print("Prediksi Salah : ", prediksiSalah, "data")
print("Akurasi :", prediksiBenar/(prediksiBenar+prediksiSalah) * 100, "%")
```

```
Prediksi Benar :  4 data
Prediksi Salah :  16 data
Akurasi : 20.0 %
```

Di akhir, dilakukan input ‘prediksiBenar’ dan ‘prediksiSalah’ dari data dimana dalam mencari akurasi diperlukan formula ‘prediksiBenar’ dibagi dengan ‘(prediksiBenar+prediksiSalah’) dikali 100% yang menghasilkan output prediksi Benar sejumlah 4 data, prediksi salah 16 data dan akurasi sebesar 20%.

4. Kesimpulan

Berdasarkan hasil yang kami uji coba, didapatkan kesimpulan bahwa Algoritma Decision Tree dalam Menganalisis Prediksi Penyakit Jantung dengan kesimpulan bahwa terdapat 4 data dengan prediksi benar, 16 data prediksi salah dengan hasil akurasi akhir sebesar 20%.

Referensi

Sellappan Palaniappan & Rafiah Awang, "Intelligent Heart Disease Prediction System Using Data Mining Techniques," IJCSNS International Journal of Computer Science and Network Security, vol. 8, Agustus 2008.

Rohman, A., 2013. Penerapan algoritma c4. 5 berbasis adaboost untuk prediksi penyakit jantung. Dinamika Sains, 11(26).

Mardi, Y. (2017). Data Mining: Klasifikasi Menggunakan Algoritma C4. 5. Jurnal Edik Informatika Penelitian Bidang Komputer Sains dan Pendidikan Informatika, 2(2), 213-219.

Lampiran

https://colab.research.google.com/drive/1-LcMHnCIH4YQXKKfjau55OJ2ODczLGvO?usp=sharing#scrollTo=dDlgeS5Z5spj