LAPORAN TUGAS BESAR 1B
IF3270 Pembelajaran Mesin
IMPLEMENTASI DECISION TREE LEARNING

Disusun oleh:

Lukas Kurnia Jonathan / 13517006
Eginata Kasan / 13517030
Vivianni / 13517060
Rika Dewi / 13517147

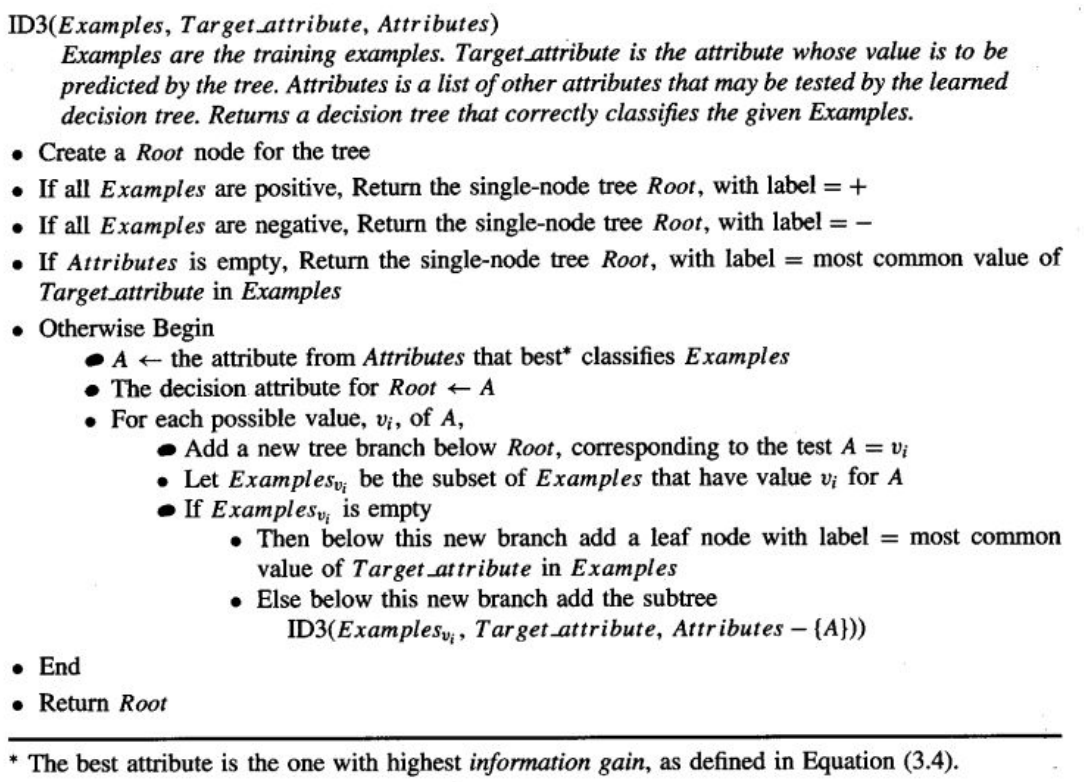
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2020

Daftar Isi

Bab 1 Implementasi

Pada tugas besar kali ini, kami melakukan implementasi modul myID3 dan myC45 untuk menangani isu-isu DTL dengan menggunakan *Python.*



ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A}))
- End
- Return *Root*

* The best attribute is the one with highest *information gain*, as defined in Equation (3.4).

Gambar 1. Algoritma modul ID3 (sumber: Machine Learning, Tom Mitchell)

Pembangunan *Decision Tree* dilakukan dengan mengikuti aturan sebagai berikut untuk mengatasi isu-isu DTL:

1. Terdapat *post-pruning* dengan 20% dari *training data* digunakan sebagai *testing data* untuk menghindari terjadinya *overfitting*
2. Penanganan atribut kontinu sebagai pivot dengan menggunakan information gain dari kandidat
3. Terdapat mode pemilihan atribut alternatif yaitu dengan menggunakan *gain ratio*
4. Penggunaan *most common target value* untuk menangani *missing value*

1.1 Post Pruning

Pruning pada modul ini diawali dengan mengkonversikan *decision tree* menjadi *list of rules*. Sebuah rule terdiri atas kondisi dan target yang ditampilkan dalam wujud berupa list seperti sebagai berikut:

[*condition1, condition2, …, condition n, target value*]

Setiap rule dalam *List of rules* kemudian dicari *superset*nya dan akan dihitung akurasi rule untuk setiap elemen (kondisi) dari *superset of rules*. Elemen dari superset yang memiliki akurasi tertinggi akan dipilih untuk menggantikan rule yang lama, atau dengan kata lain, kondisi-kondisi yang menurunkan akurasi dari branch tersebut akan dipangkas dari *Decision Tree*. Setelah semua rule berhasil dipangkas, *List of Rules* yang baru akan diurutkan menurun berdasarkan akurasi tertinggi.

1.2 Continuous Value

Penanganan atribut dengan nilai kontinu dilakukan dengan mengubah atribut tersebut menjadi dua buah atribut diskrit dengan nilai HIGH dan LOW. Penanganan dilakukan dengan mencari *threshold* yang memiliki nilai information gain tertinggi, dimana *threshold* adalah data training yang memiliki perbedaan target value. Mula-mula data kontinu diurutkan dari kecil ke besar kemudian dilakukan identifikasi seluruh threshold yang ada dan dicari information gain dari kandidat *threshold*. Setelah information gain terbesar didapatkan, value dengan nilai lebih kecil dari *threshold* dibuat diskrit dengan value LOW, sedangkan value dengan nilai yang lebih besar sama dengan diberi nilai HIGH. Proses pembelajaran dilanjutkan seperti halnya menangani data yang diskrit.

1.3 Gain Ratio

Gain Ratio adalah mode pemilihan atribut alternatif selain *Information Gain*. Pemilihan mode dapat dilakukan ketika memanggil fungsi *build* untuk *Decision Tree* dengan memasukkan parameter *True* pada *isGainRatio*. *Gain Ratio* didapatkan dengan membagi *Information Gain* dengan *Entropy.*

1.4 Handling Missing Value

Setelah membaca dataframe yang diberikan, akan dilakukan pencarian kolom yang memiliki value yang kosong atau NaN (Not a Number). Kemudian, value yang paling sering muncul (modus) dari kolom tersebut akan digunakan untuk mengisi data yang bernilai NaN. Proses ini dilakukan hingga tidak terdapat lagi data yang bernilai NaN pada dataframe.

Bab 2 Hasil Eksekusi

2.1 Dataset play-tennis

- Hasil menggunakan myID3 untuk full-training, data diskrit:

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
'outlook'
|--'humidity' (sunny)
   |--'no' (high)
   |--'yes' (normal)
|--'yes' (overcast)
|--'windy' (rainy)
   |--'yes' (False)
   |--'no' (True)
```

- Hasil menggunakan myID3 untuk full-training, data diskrit dan isu C45 menggunakan gainRatio:

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
'outlook'
|--'humidity' (sunny)
   |--'no' (high)
   |--'yes' (normal)
|--'yes' (overcast)
|--'windy' (rainy)
   |--'yes' (False)
   |--'no' (True)
```

- Isu C45 handling missing value, dilakukan dengan mencari most common value dalam sebuah feature. (contoh: most common adalah *mild*):

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
     outlook  temp humidity  windy play
0      sunny   hot     high  False   no
1      sunny   hot     high   True   no
2   overcast   hot     high  False  yes
3      rainy  mild     high  False  yes
4      rainy   NaN   normal  False  yes
5      rainy  cool   normal   True   no
6   overcast  cool   normal   True  yes
7      sunny  mild     high  False   no
8      sunny  cool   normal  False  yes
9      rainy  mild   normal  False  yes
10     sunny  mild   normal   True  yes
11  overcast  mild     high   True  yes
12  overcast   hot   normal  False  yes
13     rainy  mild     high   True   no
```

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
     outlook  temp humidity  windy play
0      sunny   hot     high  False   no
1      sunny   hot     high   True   no
2   overcast   hot     high  False  yes
3      rainy  mild     high  False  yes
4      rainy  mild   normal  False  yes
5      rainy  cool   normal   True   no
6   overcast  cool   normal   True  yes
7      sunny  mild     high  False   no
8      sunny  cool   normal  False  yes
9      rainy  mild   normal  False  yes
10     sunny  mild   normal   True  yes
11  overcast  mild     high   True  yes
12  overcast   hot   normal  False  yes
13     rainy  mild     high   True   no
```

- Hasil menggunakan post-pruning (80% training dan 20% testing) dengan menghasilkan list of rule. Dibaca dari depan ke belakang menurut urutan akurasi (akurasi rule[0] > akurasi rule[1]).

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
'outlook'
|--'humidity' (sunny)
   |--'no' (high)
   |--'yes' (normal)
|--'yes' (overcast)
|--'windy' (rainy)
   |--'yes' (False)
   |--'no' (True)

[[outlook= overcast, 'yes'], [outlook= rainy, windy= True, 'no'], [outlook= sunny, humidity= high, 'no'], [o
utlook= sunny, humidity= normal, 'yes'], [outlook= rainy, windy= False, 'yes']]
```

2.2 Dataset iris

- Hasil menggunakan myID3 untuk full-training, data kontinu:

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
'petal_length'
|--'setosa' (< 2.45)
|--'petal_width' (>= 2.45)
   |--'sepal_length' (< 1.75)
      |--'sepal_width' (< 7.1)
         |--'versicolor' (< 2.85)
         |--'versicolor' (>= 2.85)
      |--'virginica' (>= 7.1)
   |--'sepal_width' (>= 1.75)
      |--'virginica' (< 3.15)
      |--'sepal_length' (>= 3.15)
         |--'versicolor' (< 6.05)
         |--'virginica' (>= 6.05)
```

- Hasil menggunakan myID3 untuk full-training, data kontinu dan isu C45 menggunakan gainRatio:

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
'petal_length'
|--'setosa' (< 2.45)
|--'petal_width' (>= 2.45)
   |--'sepal_length' (< 1.75)
      |--'sepal_width' (< 7.1)
         |--'versicolor' (< 2.85)
         |--'versicolor' (>= 2.85)
      |--'virginica' (>= 7.1)
   |--'sepal_width' (>= 1.75)
      |--'virginica' (< 3.15)
      |--'sepal_length' (>= 3.15)
         |--'versicolor' (< 6.05)
         |--'virginica' (>= 6.05)
```

- Isu C45 handling missing value, dilakukan dengan mencari most common value dalam sebuah feature:

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          NaN           1.4          0.2     setosa
1             4.9          3.0           NaN          0.2     setosa
2             4.7          3.2           1.3          NaN     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
5             5.4          3.9           1.7          0.4     setosa
6             4.6          3.4           1.4          0.3     setosa
```

```
(base) lukaskurnia@Ipray-J22:~/Lukas/Kuliah/Program/ML/Decision-Tree-Learning$ python -W ignore main.py
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.0           1.4          0.2     setosa
1             4.9          3.0           1.5          0.2     setosa
2             4.7          3.0           1.5          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
5             5.4          3.9           1.7          0.4     setosa
6             4.6          3.4           1.4          0.3     setosa
7             5.0          3.4           1.5          0.2     setosa
```

- Hasil menggunakan post-pruning (80% training dan 20% testing) dengan menghasilkan list of rule. Dibaca dari depan ke belakang menurut urutan akurasi (akurasi rule[0] > akurasi rule[1]).

```
'petal_length'
|--'setosa' (< 2.45)
|--'petal_width' (>= 2.45)
   |--'sepal_length' (< 1.65)
      |--'versicolor' (< 6.0)
      |--'sepal_width' (>= 6.0)
         |--'setosa' (< 2.2)
         |--'versicolor' (>= 2.2)
   |--'sepal_width' (>= 1.65)
      |--'virginica' (< 3.0)
      |--'sepal_length' (>= 3.0)
         |--'versicolor' (< 6.1)
         |--'virginica' (>= 6.1)

[[petal_length= < 2.45, 'setosa'], [petal_length= >= 2.45, petal_width= < 1.65, sepal_length= < 6.0, 'versic
olor'], [petal_length= >= 2.45, petal_width= < 1.65, sepal_length= >= 6.0, sepal_width= < 2.2, 'setosa'], [p
etal_length= >= 2.45, petal_width= < 1.65, sepal_length= >= 6.0, sepal_width= >= 2.2, 'versicolor'], [petal_
length= >= 2.45, petal_width= >= 1.65, sepal_width= < 3.0, 'virginica'], [petal_length= >= 2.45, petal_width
= >= 1.65, sepal_width= >= 3.0, sepal_length= < 6.1, 'versicolor'], [petal_length= >= 2.45, petal_width= >=
```

Bab 3 Perbandingan dengan Library

3.1 Perbandingan dengan hasil DTL sklearn

Dataset play-tennis:

```
|--- outlook <= 0.50
|   |--- class: 1
|--- outlook >  0.50
|   |--- humidity <= 0.50
|   |   |--- outlook <= 1.50
|   |   |   |--- windy <= 0.50
|   |   |   |   |--- class: 1
|   |   |   |--- windy >  0.50
|   |   |   |   |--- class: 0
|   |   |--- outlook >  1.50
|   |   |   |--- class: 0
|   |--- humidity >  0.50
|   |   |--- windy <= 0.50
|   |   |   |--- class: 1
|   |   |--- windy >  0.50
|   |   |   |--- temp <= 1.00
|   |   |   |   |--- class: 0
|   |   |   |--- temp >  1.00
|   |   |   |   |--- class: 1
```

Pembentukan *Decision Tree Learning* yang dilakukan oleh library sklearn berbeda dengan yang diimplementasikan dimana pada library sklearn, semua value yang kategorikal akan dijadikan kontinu terlebih dahulu dan kemudian dilakukan pembentukan *decision tree* sedangkan pada hasil implementasi kami, data kategorikal tetap diproses tanpa mengubah value dari data tersebut.

Dataset iris:

```
|--- petal width (cm) <= 0.80
|   |--- class: 0
|--- petal width (cm) >  0.80
|   |--- petal width (cm) <= 1.75
|   |   |--- petal length (cm) <= 4.95
|   |   |   |--- petal width (cm) <= 1.65
|   |   |   |   |--- class: 1
|   |   |   |--- petal width (cm) >  1.65
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.95
|   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |--- class: 2
|   |   |   |--- petal width (cm) >  1.55
|   |   |   |   |--- petal length (cm) <= 5.45
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- petal length (cm) >  5.45
|   |   |   |   |   |--- class: 2
|   |--- petal width (cm) >  1.75
|   |   |--- petal length (cm) <= 4.85
|   |   |   |--- sepal length (cm) <= 5.95
|   |   |   |   |--- class: 1
|   |   |   |--- sepal length (cm) >  5.95
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.85
|   |   |   |--- class: 2
```

Pembentukan *tree* pada *library* DTL dan pada *library* ID3 pada dasarnya memiliki konsep dan algoritma yang berbeda sehingga *tree* yang dihasilkan oleh algoritma DTL berbeda dengan tree yang diimplementasikan dan juga tree yang dihasilkan oleh library Id3Estimator.

Library DecisionTreeClassifier menggunakan algoritma minimal CCP (*Cost Complexity Pruning*) untuk melakukan pruning. Pada algoritma ini terdapat parameter alpha yang disebut sebagai complexity parameter. Algoritma CCP mencari sub-pohon dari pohon T yang meminimalkan cost-complexity measure. Pada node tersebut akan dicari link terlemah untuk dipruning. Link terlemah ini didapat dari nilai non-leaf node dengan nilai alpha efektif paling kecil. Hasil pruning dengan library ini adalah sebuah tree, sementara hasil pruning DTL implementasi merupakan sebuah *list of rules* yang apabila dikonversi kembali menjadi tree akan berbeda dari tree yang asli dan dapat terdiri atas banyak tree baru yang memiliki root yang berbeda.

3.2 Perbandingan dengan hasil DTL Id3Estimator

Dataset play-tennis:

```
outlook <=0.50: 1 (4)
outlook >0.50
|   humidity <=0.50
|   |   temp <=1.50: 0 (2)
|   |   temp >1.50
|   |   |   windy <=0.50: 0 (1/1)
|   |   |   windy >0.50: 0 (1)
|   humidity >0.50
|   |   windy <=0.50: 1 (3)
|   |   windy >0.50
|   |   |   temp <=1.00: 0 (1)
|   |   |   temp >1.00: 1 (1)
```

Pada *library Id3Estimator*, *value* yang dapat diproses hanyalah *value* yang bernilai kontinu, sehingga sebelum melakukan pembentukan *tree*, semua *value* yang bersifat kategorikal diproses dahulu hingga menjadi bentuk kontinu. Pada algoritma yang diimplementasikan, algoritma dapat memproses *value* yang bersifat atribut sehingga cara pembentukan *tree* pada algoritma yang diimplementasikan berbeda dengan *library Id3Estimator.*

Dataset iris:

```
petal length (cm) <=2.45: 0 (50)
petal length (cm) >2.45
|   petal width (cm) <=1.75
|   |   sepal length (cm) <=7.10
|   |   |   sepal width (cm) <=2.85: 1 (27/4)
|   |   |   sepal width (cm) >2.85: 1 (22)
|   |   sepal length (cm) >7.10: 2 (1)
|   petal width (cm) >1.75
|   |   sepal length (cm) <=5.95
|   |   |   sepal width (cm) <=3.10: 2 (6)
|   |   |   sepal width (cm) >3.10: 1 (1)
|   |   sepal length (cm) >5.95: 2 (39)
```

Hasil *tree* yang dibentuk baik oleh *tree* yang diimplementasikan maupun dari *library Id3Estimator* memiliki bentuk yang serupa dan nilai yang hampir sama. Perbedaan nilai ini mungkin terjadi ketika dilakukan pembulatan ketika menghitung entropy dan *information gain*.

Pruning pada *Id3Estimator* juga berbeda dengan DTL implementasi. Pada *Id3Estimator*, hasil pruning merupakan sebuah *tree* yang dipangkas mulai dari node paling bawah sehingga membentuk *tree* yang mempertahankan *root*nya, sedangkan pada DTL hasil implementasi, hasil pruning adalah *list of rules* yang apabila dikonversi kembali menjadi tree akan berbeda dari tree yang asli dan dapat terdiri atas banyak tree baru yang memiliki root yang berbeda. Terdapat juga perbedaan pada persentase *splitting data* untuk testing dan training, yaitu pada *Id3Estimator*, 30% digunakan untuk hasil testing, sementara persentase pada DTL hasil implementasi yang digunakan adalah 20%.

Bab 4 Pembagian Tugas

Nama - NIM	Tugas
Lukas Kurnia J. 13517006	Continuous Value, Alternatives measure for selecting attributes (Gain Ratio), Laporan
Eginata Kasan 13517030	Pruning, Laporan
Vivianni 13517060	Handling Missing Value, Rumus Gain dan Entropy, Laporan
Rika Dewi 13517147	Struktur Data Tree, Main, Splitting