

**Tugas Besar 1B**  
**IF3270 - Pembelajaran Mesin**  
**Implementasi Decision Tree Learning**



Tanggal Penyusunan :  
14 Februari 2020

Penyusun :

Irfan Haris W	13517041
Marsa Thoriq Ahmada	13517071
Mgs Riandi Ramadhan	13517080
Fajar Muslim	13517149

**Teknik Informatika**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**2019**

## A. Penjelasan implementasi

Algoritma Id3Estimator yang kita implementasikan mengacu pada buku Machine Learning Tom Mitchell. Dalam membuat implementasi tersebut kami menggunakan beberapa library dari python yaitu :

- sklearn.datasets  
Digunakan untuk load data iris
- pandas  
Digunakan untuk membaca dataset play\_tennis dan memanipulasi data set
- numpy  
Digunakan untuk memeriksa tipe data
- math  
Digunakan untuk menghitung logaritma
- copy  
Digunakan untuk menangani pass by reference. Copy objek dengan membuat objek baru yang memiliki address yang berbeda dengan objek awalnya

Selain itu, kami juga membuat satu kelas dan beberapa fungsi yang kita bangun yaitu :

- Kelas Node  
Kelas node ini digunakan dalam membentuk model yang berupa tree. Kelas ini menyimpan informasi berupa atribut, label, vertex parent, most common label, dan node node anaknya. Kelas node di instansiasi saat membentuk model yang berupa tree, baik itu Id3Estimator maupun C45.
- Fungsi get\_most\_common\_label  
Mengembalikan suatu nilai atribut yang memiliki nilai kemunculan paling banyak
- Fungsi information\_gain  
Menghitung information gain pada atribut tertentu
- Fungsi entropy  
Menghitung entropy pada atribut tertentu
- Fungsi gain\_ratio  
Menghitung gain\_ratio
- Fungsi print\_tree  
Mencetak model tree yang telah terbentuk
- Fungsi copy\_tree

Meng-copy tree. Fungsi ini digunakan untuk menangani post pruning. Karena pada saat dilakukan pruning harus dilakukan penghapusan node untuk temp\_tree yang mirip dengan tree yang sebelumnya (belum di pruning). Dalam kasus ini python menggunakan pass by reference. Sehingga kita perlu membuat tree baru hasil pruning yang mempunyai address yang berbeda dengan node sebelumnya.

- Fungsi check\_tree  
Mendapatkan hasil prediksi untuk setiap example data test (1 row) sesuai tree yang telah dibentuk
- Fungsi pred  
Mengembalikan prediksi untuk satu data test
- Fungsi accuracy  
Menghitung accuracy hasil prediksi
- Fungsi get\_data\_validate  
Melakukan split data untuk digunakan sebagai validasi, sebesar 20 persen dari dataset total
- Fungsi prune\_tree  
Menghapus node tertentu
- Fungsi post\_pruning  
Melakukan pruning pada node yang tidak mengubah akurasi model
- Fungsi split\_information  
Menghitung split information untuk atribut tertentu

-> Algoritma Id3Estimator yang digunakan

**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*

-> C45 dibangun dengan menambahkan beberapa fungsional di Id3Estimator seperti :  
penanganan untuk atribut kontinu dan post pruning

-> Petunjuk Menjalankan program

Jalankan script berikut pada direktori program :

python3 main.py

## B. Hasil eksekusi

### My C45 dataset iris

```
[petal length (cm)]
---->=2.45
    [petal width (cm)]
    ----->=1.75
        2
    -----<1.75
        [petal length (cm)]
        ----->=4.95
            [petal width (cm)]
            ----->=1.55
                [sepal length (cm)]
                ----->=6.95
                    2
                -----<6.95
                    1
            -----<1.55
                2
        -----<4.95
            [petal width (cm)]
            ----->=1.65
                2
            -----<1.65
                1
        ----<2.45
            0
```

### My Id3Estimator dataset iris

Contoh gambar ini hanya sebagian dari tree yang terbentuk. Implementasi algoritma Id3Estimator kami sama dengan Id3Estimator di buku Machine Learning Tom Mitchel. Pada buku itu disebutkan bahwa Id3Estimator memang belum bisa menangani continues value. Continues value baru ditangani pada algoritma decision tree C45 yang merupakan Id3Estimator

```
[petal length (cm)]
----1.4
    0
----1.3
    0
----1.5
    0
----1.7
    0
----1.6
    0
----1.1
    0
----1.2
    0
----1.0
    0
----1.9
    0
----4.7
    1
----4.5
    [sepal length (cm)]
    -----6.4
        1
    -----5.7
        1
    -----5.6
        1
    -----6.2
```

## My C45 dataset play\_tennis

```
[outlook]
----Sunny
      No
----Overcast
      Yes
----Rain
      Yes
```

## My Id3Estimator dataset play\_tennis

```
[outlook]
----Sunny
      [humidity]
      -----High
            No
      -----Normal
            Yes
----Overcast
      Yes
----Rain
      [wind]
      -----Weak
            Yes
      -----Strong
            No
```

### C. Perbandingan dengan hasil DTL sklearn dan Id3Estimator

#### DTL sklearn 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
|                       |--- sepal length (cm) <= 6.95
|                           |--- class: 1
|                           |--- sepal length (cm) > 6.95
|                               |--- class: 2
|       |--- petal width (cm) > 1.75
|           |--- petal length (cm) <= 4.85
|               |--- sepal width (cm) <= 3.10
|                   |--- class: 2
|                   |--- sepal width (cm) > 3.10
|                       |--- class: 1
|           |--- petal length (cm) > 4.85
|               |--- class: 2
```

## My C45 dataset iris

```
[petal length (cm)]
---->=2.45
    [petal width (cm)]
    ----->=1.75
        2
    -----<1.75
        [petal length (cm)]
        ----->=4.95
            [petal width (cm)]
            ----->=1.55
                [sepal length (cm)]
                ----->=6.95
                    2
                -----<6.95
                    1
            -----<1.55
                2
        -----<4.95
            [petal width (cm)]
            ----->=1.65
                2
            -----<1.65
                1
    ----<2.45
        0
```

## 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)
```

## My Id3Estimator dataset iris

Contoh gambar ini hanya sebagian dari tree yang terbentuk. Implementasi algoritma Id3Estimator kami sama dengan Id3Estimator di buku Machine Learning Tom Mitchel. Pada buku itu disebutkan bahwa Id3Estimator memang belum bisa menangani continues value. Continues value baru ditangani pada algoritma decision tree C45 yang merupakan Id3Estimator



```

[petal length (cm)]
----1.4
      0
----1.3
      0
----1.5
      0
----1.7
      0
----1.6
      0
----1.1
      0
----1.2
      0
----1.0
      0
----1.9
      0
----4.7
      1
----4.5
      [sepal length (cm)]
-----6.4
          1
-----5.7
          1
-----5.6
          1
-----6.2

```

### DTL sklearn dataset play\_tennis

```

|--- outlook <= 0.50
|   |--- class: Yes
|--- outlook > 0.50
|   |--- humidity <= 0.50
|       |--- outlook <= 1.50
|           |--- wind <= 0.50
|               |--- class: No
|               |--- wind > 0.50
|                   |--- class: Yes
|           |--- outlook > 1.50
|               |--- class: No
|       |--- humidity > 0.50
|           |--- wind <= 0.50
|               |--- outlook <= 1.50
|                   |--- class: No
|                   |--- outlook > 1.50
|                       |--- class: Yes
|       |--- wind > 0.50
|           |--- class: Yes

```

### My C45 dataset play\_tennis

```

[outlook]
----Sunny
      No
----Overcast
      Yes
----Rain
      Yes

```

## Id3Estimator dataset play\_tennis

```
outlook <=0.50: Yes (4)
outlook >0.50
|  humidity <=0.50
|  |  day <=7.00: No (3)
|  |  day >7.00: No (1/1)
|  humidity >0.50
|  |  day <=9.50: Yes (3)
|  |  day >9.50
|  |  |  wind <=0.50: No (1)
|  |  |  wind >0.50: Yes (1)
```

## My Id3Estimator play\_tennis

```
[outlook]
----Sunny
    [humidity]
    -----High
    No
    -----Normal
    Yes
    ----Overcast
    Yes
    ----Rain
    [wind]
    -----Weak
    Yes
    -----Strong
    No
```

#### D. Pembagian tugas setiap anggota kelompok

Pengerjaan tugas besar ini sebagian besar dilakukan saat mengerjakan bersama sama. Pembagian tugasnya dilakukan di tempat. Saat membutuhkan suatu fungsi, kita assign kepada salah satu anggota kelompok. Jika menemui kesulitan, maka kami melakukan pair programming. Kemudian jika, tugas itu belum bisa diselesaikan berdua, maka kita mendiskusikan secara bersama-sama. Kurang lebih tugas yang di assign ke masing masing anggota kelompok sebagai berikut :

NIM	Nama	Tugas	Presentase
Irfan Haris W	13517041	Rule post pruning, print_tree, copy_tree, best_atribut	25 %
Marsa Thoriq Ahmada	13517071	Accuracy, predict, handling continues value, combine function into C45	25 %
Mgs Riandi Ramadhan	13517080	Node class, rule post pruning, information gain, entropy	25 %
Fajar Muslim	13517149	Handling missing value, gain ratio, splitting data, most_comon_value	25 %