Implementasi Classifier Weka *myID3*

Tugas Pra-Praktikum IF4071

Semester I Tahun 2014 / 2015

Kelompok WbTeladan

oleh :

Iskandar Setiadi / 13511073

Raden Fajar Hadria Putra / 13511076

M Andri Eka Fauzy / 13511088

Sekolah Teknik Elektro dan Informatika (STEI ITB)

Institut Teknologi Bandung

Jl. Ganesha No. 10, Bandung 40132

Tahun 2014

A. Struktur Data

Nama Kelas : myID3
Parent Class : Classifier
Attribut :

Nama Atribut	Tipe Atribut	Fungsi Atribut
EMPTY_VALUE	double	Nilai apabila pada pohon tidak terdapat instance
node_attribute	Attribute	Atribut yang digunakan untuk *splitting*
attribute_title	array of String	Nama-nama atribut dari data yang dibuat ID3
classification	double	Nilai dari node jika node merupakan daun
node_value	Attribute	Atribut node dari dataset
isChecked	boolean	Boolean apakah Instances yang diberikan berjumlah 0 atau tidak
next_node	array of myID3	Suksesor dari node
IG	array of double	Nilai *information gain* dari semua attribute di node suatu tree.

Metode :

Nama Metode	Output Metode	Parameter	Fungsi Metode
buildClassifier	-	arg0 (Instances)	Membangun *classifier* myID3
process	-	arg0 (Instances)	Membangun pohon (*tree*) myID3
entropyCalculation	double	arg0 (Instances)	Menghitung entropi dari dataset
ntr	array of Instances	arg0 (Instance) a (Attribute)	Memisahkan dan mengelompokkan data berdasarkan atribut pada parameter
classifyInstance	double	arg0 (Instances)	Mengembalikan nilai *classValue* hasil klasifikasi

			menggunakan myID3. Selain itu, menampilkan *path* dari proses klasifikasi
toString	String	-	Menampilkan model *decision tree*
recursive_print	String	level (Integer)	Menampilkan pohon pada level tertentu

B. Implementasi

Implementasi Pembuatan Pohon :

```
@SuppressWarnings("rawtypes")
    private void process(Instances arg0) throws Exception {
        if (arg0.numInstances() == 0) {
                classification = EMPTY_VALUE;
                node_value = null;
                isChecked = true;
                return;
        } else {
                IG = new double[arg0.numAttributes()];

                Enumeration e = arg0.enumerateAttributes();
                while (e.hasMoreElements()) {
                        double information_gain = entropyCalculation(arg0);
                        Attribute att = (Attribute) e.nextElement();
                        Instances[] s = ntr(arg0, att);
                        int total_instances = arg0.numInstances();

                        for (int i = 0; i < att.numValues(); i++) {
                                int n = s[i].numInstances();
                                if (n > 0) {
                                        information_gain -= ((double) n / (double)
total_instances)
                                                        * entropyCalculation(s[i]);
                                }
                        }
                        /** Information Gain */
                        IG[att.index()] = information_gain;
                }

                int max_index = 0;
                for (int i = 1; i < arg0.numAttributes(); i++) {
                        if (IG[i] > IG[max_index]) {
                                max_index = i;
                        }
                }
                node_value = arg0.attribute(max_index);

                if (IG[node_value.index()] == 0) {
                        /** Leaf */
                        double[] num_of_elements = new double[arg0.numClasses()];
                        isChecked = true;
                        node_value = null;

                        Enumeration e2 = arg0.enumerateInstances();
                        while (e2.hasMoreElements()) {
                                Instance inst = (Instance) e2.nextElement();
                                int class_value = (int) inst.classValue();
                                num_of_elements[class_value]++;
                        }
                        Utils.normalize(num_of_elements);

                        int max_index2 = 0;
                        for (int i = 1; i < arg0.numClasses(); i++) {
                                if (num_of_elements[i] > num_of_elements[max_index]) {
                                        max_index = i;
                                }
                        }

                        classification = num_of_elements[max_index2];
                } else {
                        /** Recursive */
                        next_node = new myID3[node_value.numValues()];
                        Instances[] s = ntr(arg0, node_value);
                        for (int i = 0; i < node_value.numValues(); i++) {
                                next_node[i] = new myID3();
                                next_node[i].process(s[i]);
                        }
                }

        }
    }
```

Implementasi Penghitungan Entropi :

```
@SuppressWarnings("rawtypes")
    private double entropyCalculation(Instances arg0) throws Exception {
        double return_value = 0;

        for (int i = 0; i < arg0.numClasses(); i++) {
                int num_of_elements = 0;

                Enumeration e = arg0.enumerateInstances();
                while (e.hasMoreElements()) {
                        Instance inst = (Instance) e.nextElement();
                        if (inst.classValue() == i) {
                                num_of_elements++;
                        }
                }

                if (num_of_elements > 0) {
                        double v = (double) num_of_elements / arg0.numInstances();
                        return_value -= Utils.log2(v) * v;
                }
        }
        return return_value;
    }
```

Implementasi Penghitungan Information Gain :

```
        for (int i = 0; i < att.numValues(); i++) {
                                int n = s[i].numInstances();
                                if (n > 0) {
                                        information_gain -= ((double) n / (double)
total_instances)
                                                        * entropyCalculation(s[i]);
                                }
                        }
```

C. Pengujian

Data Set yang Digunakan :

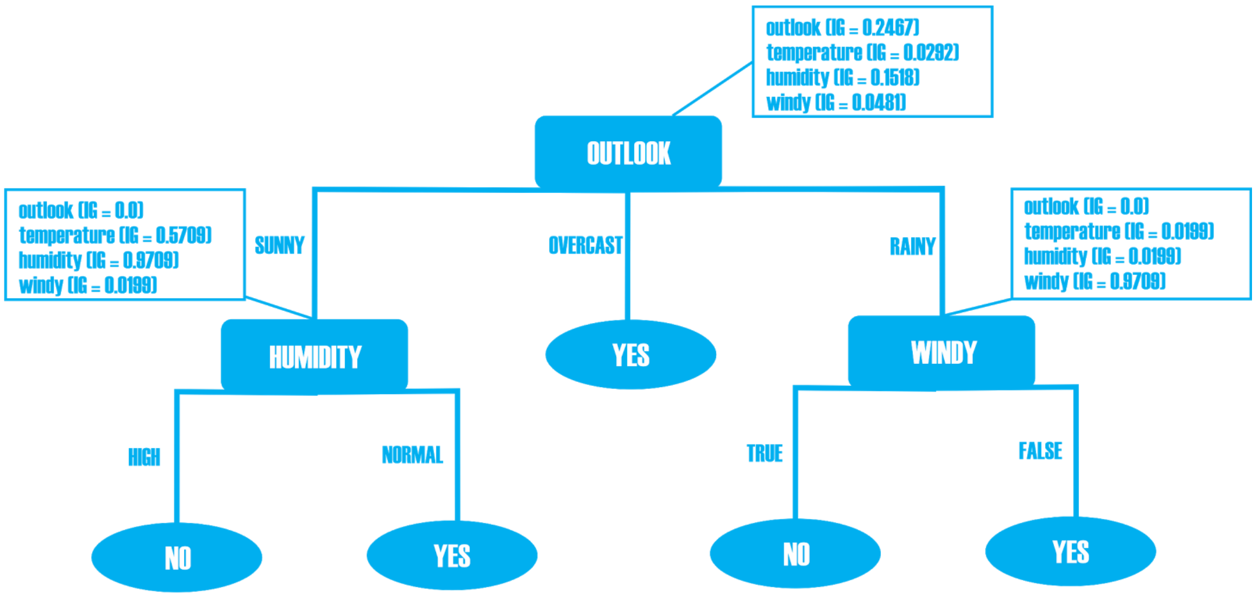
```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

weather.nominal.arff

ID3 yang dihasilkan :

```
Id3
|
 outlook (IG = 0.24674981977443894)
 temperature (IG = 0.029222565658954536)
 humidity (IG = 0.15183550136234125)
 windy (IG = 0.04812703040826921)
 outlook = sunny

|
|
 outlook (IG = 0.0)
 temperature (IG = 0.5709505944546683)
 humidity (IG = 0.9709505944546684)
 windy (IG = 0.019973094021974558)
 humidity = high
 Kelas : no [LEAF]

|
|
 outlook (IG = 0.0)
 temperature (IG = 0.5709505944546683)
 humidity (IG = 0.9709505944546684)
 windy (IG = 0.019973094021974558)
 humidity = normal
 Kelas : yes [LEAF]



|
 outlook (IG = 0.24674981977443894)
 temperature (IG = 0.029222565658954536)
 humidity (IG = 0.15183550136234125)
 windy (IG = 0.04812703040826921)
 outlook = overcast
 Kelas : yes [LEAF]

|
 outlook (IG = 0.24674981977443894)
 temperature (IG = 0.029222565658954536)
 humidity (IG = 0.15183550136234125)
 windy (IG = 0.04812703040826921)
 outlook = rainy

|
|
 outlook (IG = 0.0)
 temperature (IG = 0.019973094021974558)
 humidity (IG = 0.019973094021974558)
 windy (IG = 0.9709505944546684)
 windy = TRUE
 Kelas : no [LEAF]

|
|
 outlook (IG = 0.0)
 temperature (IG = 0.019973094021974558)
 humidity (IG = 0.019973094021974558)
 windy (IG = 0.9709505944546684)
 windy = FALSE
 Kelas : yes [LEAF]
```

Ilustrasi Pohon :


outlook (IG = 0.2467)
temperature (IG = 0.0292)
humidity (IG = 0.1518)
windy (IG = 0.0481)
OUTLOOK
outlook (IG = 0.0)
temperature (IG = 0.5709)
humidity (IG = 0.9709)
windy (IG = 0.0199)
SUNNY
OVERCAST
RAINY
outlook (IG = 0.0)
temperature (IG = 0.0199)
humidity (IG = 0.0199)
windy (IG = 0.9709)
HUMIDITY
YES
WINDY
HIGH
NORMAL
TRUE
FALSE
NO
YES
NO
YES