Eksplorasi Waikato Environment for Knowledge Analysis (WEKA)

LAPORAN

Diajukan untuk memenuhi tugas kecil 2 mata kuliah IF3170 Intelejensi Buatan

oleh

Malvin Juanda **13514044**

Steffi Indrayani **13514063**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2016

I. HASIL PENGUJIAN IMPLEMENTASI KELAS

```
================================
  W E K A   S I M U L A T I O N
================================

Masukan Dataset : D:\Weka-3-8\data\iris.arff
Relation Name:  iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6
Num Instances:  150
Num Attributes: 5

     Name                     Type  Nom  Int Real     Missing       Unique  Dist
1 sepallength                Nom 100%   0%   0%     0 /  0%     0 /  0%     3
2 sepalwidth                 Nom 100%   0%   0%     0 /  0%     0 /  0%     3
3 petallength                Nom 100%   0%   0%     0 /  0%     0 /  0%     3
4 petalwidth                 Nom 100%   0%   0%     0 /  0%     0 /  0%     3
5 class                      Nom 100%   0%   0%     0 /  0%     0 /  0%     3

 S K E M A
1. 10-Fold Cross Validation
2. Full Training
Pilih skema :
1
J48 pruned tree
------------------

petalwidth = '(-inf-0.8]': Iris-setosa (50.0)
petalwidth = '(0.8-1.75]': Iris-versicolor (54.0/5.0)
petalwidth = '(1.75-inf)': Iris-virginica (46.0/1.0)

Number of Leaves  :        3

Size of the tree :  4

10-fold cross valdation evaluation :

Correctly Classified Instances         141               94      %
Incorrectly Classified Instances         9                6      %
Kappa statistic                          0.91
Mean absolute error                      0.0598
Root mean squared error                  0.193
Relative absolute error                 13.4523 %
Root relative squared error             40.9465 %
Total Number of Instances              150

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica

Masukan instance baru :
Nilai atribut 1 : 1
Nilai atribut 2 : 1
Nilai atribut 3 : 1
Nilai atribut 4 : 1
Hasil Klasifikasi : Iris-versicolor
```

II. PENGECEKAN KEBENARAN IMPLEMENTASI

a. Hasil Implementasi Kelas

1. Membaca dataset **iris.arff** dan mengaplikasikan filter **Discretize**

```
================================
  W E K A   S I M U L A T I O N
================================

Masukan Dataset : D:\Weka-3-8\data\iris.arff
Relation Name:  iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-
precision6
Num Instances:  150
Num Attributes: 5

     Name                    Type  Nom  Int Real     Missing       Unique  Dist
1 sepallength               Nom 100%   0%   0%     0 /  0%     0 /  0%     3
2 sepalwidth                Nom 100%   0%   0%     0 /  0%     0 /  0%     3
3 petallength               Nom 100%   0%   0%     0 /  0%     0 /  0%     3
4 petalwidth                Nom 100%   0%   0%     0 /  0%     0 /  0%     3
5 class                     Nom 100%   0%   0%     0 /  0%     0 /  0%     3
```

2. Melakukan pembelajaran dengan skema **10-fold cross validation**

```
S K E M A
1. 10-Fold Cross Validation
2. Full Training
Pilih skema :
1
J48 pruned tree
------------------

petalwidth = '(-inf-0.8]': Iris-setosa (50.0)
petalwidth = '(0.8-1.75]': Iris-versicolor (54.0/5.0)
petalwidth = '(1.75-inf)': Iris-virginica (46.0/1.0)

Number of Leaves  :        3

Size of the tree :  4

10-fold cross valdation evaluation :

Correctly Classified Instances         141               94      %
Incorrectly Classified Instances         9                6      %
Kappa statistic                          0.91
Mean absolute error                      0.0598
Root mean squared error                  0.193
Relative absolute error                 13.4523 %
Root relative squared error             40.9465 %
Total Number of Instances              150

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0   5 45 |  c = Iris-virginica
```

3. Melakukan pembelajaran dengan skema **Full Training**

```
S K E M A
1. 10-Fold Cross Validation
```

```
2. Full Training
Pilih skema :
2
J48 pruned tree
------------------

petalwidth = '(-inf-0.8]': Iris-setosa (50.0)
petalwidth = '(0.8-1.75]': Iris-versicolor (54.0/5.0)
petalwidth = '(1.75-inf)': Iris-virginica (46.0/1.0)

Number of Leaves  :        3

Size of the tree :  4

Full Training evaluation :

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.049
Root mean squared error                  0.1566
Relative absolute error                 11.0306 %
Root relative squared error             33.2123 %
Total Number of Instances              150

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

4. Membuat *instance* baru sesuai masukan dari pengguna untuk setiap nilai atribut dan melakukan klasifikasi dengan memanfaatkan model/hipotesis dan instance sesuai masukan

```
Masukan instance baru :
Nilai atribut 1 : 1
Nilai atribut 2 : 1
Nilai atribut 3 : 1
Nilai atribut 4 : 1
Hasil Klasifikasi : Iris-versicolor
```

b. Hasil Implementasi pada Weka

a. Skema 10-Fold Cross Validation

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:           iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-
precision6
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------
```

```
petalwidth = '(-inf-0.8]': Iris-setosa (50.0)
petalwidth = '(0.8-1.75]': Iris-versicolor (54.0/5.0)
petalwidth = '(1.75-inf)': Iris-virginica (46.0/1.0)

Number of Leaves  :       3

Size of the tree :        4


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         141               94       %
Incorrectly Classified Instances         9                6       %
Kappa statistic                          0.91
Mean absolute error                      0.0598
Root mean squared error                  0.193
Relative absolute error                 13.4523 %
Root relative squared error             40.9465 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall    F-Measure   MCC        ROC
Area  PRC Area  Class
                 1.000     0.000     1.000       1.000     1.000       1.000     1.000
1.000     Iris-setosa
                 0.920     0.050     0.902       0.920     0.911       0.866     0.938
0.857     Iris-versicolor
                 0.900     0.040     0.918       0.900     0.909       0.864     0.943
0.865     Iris-virginica
Weighted Avg.    0.940     0.030     0.940       0.940     0.940       0.910     0.960
0.907

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

b. Skema Full Training

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:           iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-
precision6
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
------------------

petalwidth = '(-inf-0.8]': Iris-setosa (50.0)
petalwidth = '(0.8-1.75]': Iris-versicolor (54.0/5.0)
```

```
petalwidth = '(1.75-inf)': Iris-virginica (46.0/1.0)

Number of Leaves  :       3

Size of the tree :        4


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.049
Root mean squared error                  0.1566
Relative absolute error                 11.0306 %
Root relative squared error             33.2123 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall    F-Measure   MCC        ROC
Area  PRC Area  Class
                 1.000     0.000     1.000       1.000     1.000       1.000     1.000
1.000     Iris-setosa
                 0.980     0.050     0.907       0.980     0.942       0.913     0.970
0.899     Iris-versicolor
                 0.900     0.010     0.978       0.900     0.938       0.910     0.970
0.930     Iris-virginica
Weighted Avg.    0.960     0.020     0.962       0.960     0.960       0.941     0.980
0.943

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

c. Perbandingan

Dari hasil pengecekan di atas dapat dilihat bahwa hasil dari pembelajaran dengan skema full training dari kelas implementasi sama dengan pembelajaran dengan skema full training dengan menggunakan weka. Selain itu, hasil dari pembelajaran dengan skema 10-fold cross validation kelas implementasi sama dengan pembelajaran dengan skema 10-fold cross validation weka.

III. *SOURCE CODE* KELAS IMPLEMENTASI

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package tucil_weka;
```

```
import weka.core.DenseInstance;
import weka.core.Instances;
import weka.core.Instance;
import weka.filters.Filter;
import weka.filters.supervised.attribute.Discretize;
import weka.classifiers.Evaluation;
import weka.classifiers.trees.J48;
import java.util.Random;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import weka.core.Attribute;
/**
 *
 * @author Asus
 */
public class Tucil_weka {

    /**
     * @param fileName
     * @param args the command line arguments
     * @return
     * @throws java.lang.Exception
     */

    public static Instances readFile (String fileName) throws Exception {
        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new FileReader(fileName));
        } catch (FileNotFoundException ex) {
            System.out.println("File Not Found");
        }

        Instances dataset = new Instances(reader);
        dataset.setClassIndex(dataset.numAttributes() - 1);
        reader.close();

        return dataset;
    }

    public static Instances applyDiscretize (Instances dataset) throws
Exception {
        //setup filter
        Discretize filter = new Discretize();
        filter.setInputFormat(dataset);

        //apply filter
        Instances outputTrain = Filter.useFilter(dataset, filter);

        return outputTrain;
    }

    public static J48 buildTreeClassifier (Instances dataset) throws Exception
{
        J48 tree = new J48();
        tree.buildClassifier(dataset);

        return tree;
```

```
    }

    public static Evaluation TenCrossValidation (Instances dataset, J48 tree)
throws Exception {
        Evaluation eval = new Evaluation(dataset);

        //10-fold cross validation
        int folds = 10;
        eval.crossValidateModel(tree, dataset, folds, new Random(1));

        return eval;
    }

    public static Evaluation FullSchemaTraining (Instances dataset, J48 tree)
throws Exception {
        Evaluation eval = new Evaluation(dataset);

        //FullSchemaTraining
        eval.evaluateModel(tree, dataset);

        return eval;
    }

    public static void saveModel (J48 tree) throws Exception {
        weka.core.SerializationHelper.write("tucil.model", tree);
    }

    public static J48 loadModel() throws Exception{
        J48 tree_new = (J48) weka.core.SerializationHelper.read("tucil.model");

        return tree_new;
    }

    public static Instance readInstance (Instances dataset) {

        System.out.println("Masukan instance baru :");
        DenseInstance user  = new DenseInstance(dataset.firstInstance());
        user.setDataset(dataset);
        user.setMissing(dataset.classIndex());
        for (int i = 0; i < dataset.classIndex(); i++) {
            Scanner sc = new Scanner(System.in);
            int j = i+1;
            System.out.print("Nilai atribut " + j + " : ");
            Float input = sc.nextFloat();
            user.setValue(i, input);
        }

        return user;
    }

    public static String classifyResult (J48 tree, Instance user) throws
Exception {
        double temp = tree.classifyInstance(user);
        String result = user.classAttribute().value((int) temp);

        return result;
    }

    public static void main(String[] args) throws IOException, Exception {
        //Welcome Message
        System.out.println("================================");
```

```
        System.out.println("  W E K A   S I M U L A T I O N");
        System.out.println("==================================");
        System.out.println();


        //Read dataset file
        System.out.print(" Masukan Dataset : ");
        Scanner s = new Scanner(System.in);
        String file = s.nextLine();
        Instances dataset = readFile(file);

        //Filtering
        Instances outputTrain = applyDiscretize(dataset);
        System.out.println(outputTrain.toSummaryString());

        //Pilihan Skema
        System.out.println(" S K E M A ");
        System.out.println("1. 10-Fold Cross Validation");
        System.out.println("2. Full Training");
        System.out.println("Pilih skema : ");
        int opt = s.nextInt();

        //Classifier
        J48 tree = buildTreeClassifier(outputTrain);
        System.out.println(tree);
        if (opt == 1) {
            //10-fold cross validation
            Evaluation eval = TenCrossValidation(outputTrain, tree);
            System.out.println("10-fold cross valdation evaluation : ");
            System.out.println(eval.toSummaryString());
            System.out.println(eval.toMatrixString());
        } else {
            //Full Training
            Evaluation eval = FullSchemaTraining(outputTrain, tree);
            System.out.println("Full Training evaluation : ");
            System.out.println(eval.toSummaryString());
            System.out.println(eval.toMatrixString());
        }
        //save model in external file
        saveModel(tree);

        //load model from external file
        J48 tree_new = loadModel();

        //Read Instances
        tree_new.buildClassifier(dataset);
        Instance user = readInstance(dataset);

        //Classify new instance
        System.out.println("Hasil  Klasifikasi  :  "  +  classifyResult(tree_new,
user));

          }

}
```