/\*

\* 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** javaweka;

**import** **java.io.BufferedReader**;

**import** **java.io.File**;

**import** **java.io.FileReader**;

**import** **java.util.Random**;

**import** **java.util.Scanner**;

**import** **weka.classifiers.Classifier**;

**import** **weka.classifiers.Evaluation**;

**import** **weka.classifiers.bayes.NaiveBayes**;

**import** **weka.classifiers.trees.J48**;

**import** **weka.core.Attribute**;

**import** **weka.core.DenseInstance**;

**import** **weka.core.Instance**;

**import** **weka.core.Instances**;

**import** **weka.core.converters.ArffSaver**;

**import** **weka.filters.Filter**;

**import** **weka.filters.supervised.attribute.Discretize**;

/\*\*

\*

\* @author Joshua A Kosasih, Jovian C

\*/

**public** **class** **JavaWeka** {

/\*\*

\* @param args the command line arguments

\*/

**public** **static** **void** **main**(String[] args) **throws** Exception {

//intro

System.out.println("=================================");

System.out.println("=========== JAVA WEKA ==========");

System.out.println("=================================");

System.out.println("Created by : ");

System.out.println("- Joshua Aditya Kosasih (13514012)");

System.out.println("- Jovian Christianto (13514101)\n");

//read file into instance

BufferedReader breader = **null**;

breader = **new** BufferedReader(**new** FileReader("C:/Program Files/Weka-3-8/data/iris.arff"));

Instances train = **new** Instances(breader);

train.setClassIndex(train.numAttributes() - **1**);

breader.close();

System.out.println("Log: File iris.arff loaded");

//filter

Discretize disc = **new** Discretize();

disc.setInputFormat(train);

Instances filtered = Filter.useFilter(train, disc);

filtered.setClassIndex(filtered.numAttributes() - **1**);

System.out.println("Log: Dataset filtered");

//for gui

Instances dataSet = filtered;

ArffSaver saver = **new** ArffSaver();

saver.setInstances(dataSet);

saver.setFile(**new** File("Desktop/coba.arff"));

saver.writeBatch();

Scanner input = **new** Scanner(System.in);

**int** cls = input.nextInt();

//build classifier

**if** (cls == **2**) {

J48 nbayes = **new** J48();

} **else** {

NaiveBayes nbayes = **new** NaiveBayes();

}

nbayes.buildClassifier(filtered);

//pilihan algoritma

System.out.println("\nPilihan algoritma : 1. 10-fold cross validation");

System.out.println(" 2. Full training set");

System.out.print("Input pilihan algoritma : ");

**boolean** eror = **true**;

**while** (eror) {

**int** algo = input.nextInt();

**if** (algo == **1**) { //algorithm cross validation

Evaluation eval = **new** Evaluation(filtered);

eval.crossValidateModel(nbayes, filtered, **10**, **new** Random(**1**));

eror = **false**;

System.out.println(eval.toSummaryString("\n10-fold cross validation\nResults\n=====", **true**));

System.out.println(eval.fMeasure(**1**) + " " + eval.precision(**1**) + " " + eval.recall(**1**));

} **else** **if** (algo == **2**) { //algorithm train set

Evaluation eval2 = **new** Evaluation(filtered);

eval2.evaluateModel(nbayes, filtered);

eror = **false**;

System.out.println(eval2.toSummaryString("\nFull training\nResults\n=====", **true**));

System.out.println(eval2.fMeasure(**1**) + " " + eval2.precision(**1**) + " " + eval2.recall(**1**));

} **else** {

System.out.println("Error, pilih 1 atau 2");

System.out.print("Input pilihan algoritma : ");

eror = **true**;

}

}

//save model

weka.core.SerializationHelper.write("newModel.model", nbayes);

System.out.println("Log: Model saved");

//load model

Classifier nb = (Classifier) weka.core.SerializationHelper.read("newModel.model");

System.out.println("Log: Model loaded");

//create instance from user input

System.out.println("\nCreate new instance from input");

**double**[] userdata = **new** **double**[**4**];

Scanner s = **new** Scanner(System.in);

System.out.print("Enter sepal length: ");

userdata[**0**] = s.nextFloat();

System.out.print("Enter sepal width: ");

userdata[**1**] = s.nextFloat();

System.out.print("Enter petal length: ");

userdata[**2**] = s.nextFloat();

System.out.print("Enter petal width: ");

userdata[**3**] = s.nextFloat();

Attribute att1 = train.attribute(**0**);

Attribute att2 = train.attribute(**1**);

Attribute att3 = train.attribute(**2**);

Attribute att4 = train.attribute(**3**);

//Attribute attC = train.attribute(4);

Instance userins = **new** DenseInstance(**5**);

userins.setDataset(train);

userins.setValue(att1, userdata[**0**]);

userins.setValue(att2, userdata[**1**]);

userins.setValue(att3, userdata[**2**]);

userins.setValue(att4, userdata[**3**]);

**if** (disc.input(userins)) {

Instance ufilter = disc.output();

System.out.println("Discretized instance: " + ufilter);

**double** label = nb.classifyInstance(ufilter);

System.out.println("The class is " + train.classAttribute().value((**int**) label));

}

}

}