Handout

for

One day Workshop with Hands-on Training

on

"Machine Learning and Its Applications"

Prepared and Presented
By

Dr. D. Asir Antony Gnana Singh, B.E., M.E., M.B.A., Ph.D.,
Department of CSE, Anna University,
BIT Campus, Tiruchirappalli-620 0024

at
UCE, Anna University, BIT-Campus,
Tiruchirappalli-620 0024
on
28th September 2019

Part – I

Working with WEKA Explorer

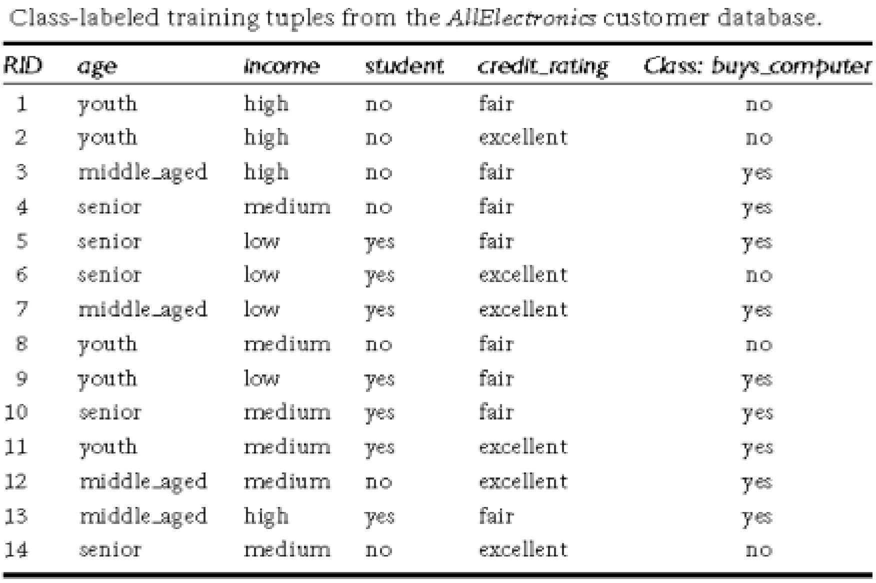
Task 1: Prediction/decision making using classification algorithm

Question 1: Predict the following unknown data/class label from the training data using tree based J48 classifier (Predict the following four customers who will buy the computer and who will not buy the computer from the historical data)

Unknown /unlabeled data:

Age	Income	Student	Credit Rating	Class (Buys Computer: Yes/No)
Youth	High	No	Fair	
Youth	High	No	Excellent	
Middle aged	High	No	Fair	
Senior	Medium	No	Fair	

Historical data / Training data:



Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Solution:

Step 1: Prepare the training dataset in arff (Attribute-Relation File Format) from the training data.

```
@relation  My_First_Training_Dataset
@attribute age {youth,middle_aged,senior}
@attribute income {high,medium,low}
@attribute student {no,yes}
@attribute credit_rating {fair,excellent}
@attribute Class {no,yes}
@data
```

```
youth,high,no,fair,no
youth,high,no,excellent,no
middle_aged,high,no,fair,yes
senior,medium,no,fair,yes
senior,low,yes,fair,yes
senior,low,yes,excellent,no
middle_aged,low,yes,excellent,yes
youth,medium,no,fair,no
youth,low,yes,fair,yes
senior,medium,yes,fair,yes
youth,medium,yes,excellent,yes
middle_aged,medium,no,excellent,yes
middle_aged,high,yes,fair,yes
senior,medium,no,excellent,no
```

Step 2: Prepare the unknown/unlabeled dataset in arff (Attribute-Relation File Format) from the unknown/unlabeled data.

```
@relation  'My_First_Unknown_Dataset'
@attribute age {youth,middle_aged,senior}
@attribute income {high,medium,low}
@attribute student {no,yes}
@attribute credit_rating {fair,excellent}
@attribute Class {no,yes}
@data
youth,high,no,fair,?
youth,high,no,excellent,?
middle_aged,high,no,fair,?
senior,medium,no,fair,?
```

Step 3: Build the predictive model using tree based j48 and save.
Step 4: Supply the unknown dataset as the test dataset.
Step 5: Choose more option and tick the output prediction option (select plain text).
Step 6: Load the model and reevaluate the model on current test set.

Question 2: Visualize the decision tree of the J48 algorithm and draw the decision tree of My_First_Training_Dataset.

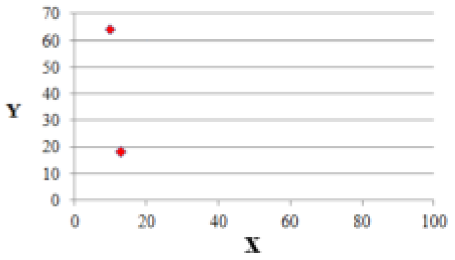
Task 2: Prediction/decision making using clustering algorithm

Question 1: Predict the following unknown data from the training data using simple K-mean clustering algorithm (predict the cluster for the given three objects).

Unknown data:

Obj. ID	X	Y	Cluster
1	13	11	
2	15	14	
3	16	18	

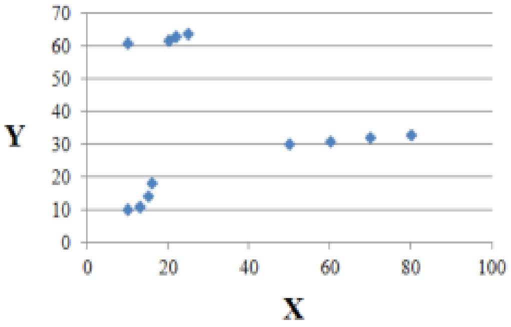
Scatter plot of unknown data



Training data:

Obj. ID	X	Y
1	10	61
2	20	62
3	22	63
4	25	64
5	50	30
6	60	31
7	70	32
8	80	33
9	10	10
10	13	11
11	15	14
12	16	18

Scatter plot of training data



Solution:

Step 1: Prepare the training dataset in arff (Attribute-Relation File Format) from the training data.

```
@relation 'My_Second_Training_Dataset'
@attribute x numeric
@attribute y numeric
@data
10,61
20,62
22,63
25,64
50,30
60,31
70,32
80,33
10,10
13,11
15,14
16,18
```

Step 2: Prepare the unknown dataset in arff (Attribute-Relation File Format) from the unknown data.

```
@relation 'My_Second_Test_Dataset'
@attribute x numeric
@attribute y numeric
@data
13,11
15,14
16,18
```

Step 3: Build the predictive model using simple K-means algorithm with 3 clusters.

Step 4: Right click on the "Result list" and click "visualize cluster assignment", and click "save" button and save the cluster assignment and view in Notepad .

Step 5: Visualize the cluster assignment for 'My_Second_Training_Dataset.arff using WEKA and fill-up the cluster number.

Obj. ID	X	Y	Cluster
1	10	61	
2	20	62	
3	22	63	
4	25	64	
5	50	30	
6	60	31	
7	70	32	
8	80	33	
9	10	10	
10	13	11	
11	15	14	
12	16	18	

Step 6: Right click on the result item for your model in the "Result list" on the "Cluster" tab. Click "Save model" from the right click menu.

Step 7: Supply the unknown dataset as the test dataset.

Step 8: Choose more option and tick the output prediction option.

Step 9: Load the model and reevaluate the model on current test set.

Step 10: Fill-up cluster number in the unknown data table for the unknown data.

Task 3: Finding the accuracy of the classification algorithms for various dataset

Question 1: Find the **accuracy** and time **taken to build the model** for the dataset **segment challenge** and encircle the **highly accurate** test option mode with **tree based J48** classifier.

Name of the Dataset	Mode of Test option	Accuracy	Time to build model
Segment-challenge	Use train set as the test dataset		
Segment-challenge	Supply **Segment-test dataset**		
Segment-challenge	Cross Validation Folds =10		
Segment-challenge	Percentage Split = 66%		

Question 2: Find the classifier which produces higher **accuracy** for the dataset diabetes with **Naive Bayes NB, tree based J48, rule based IBI** classifiers with the test option **Cross Validation Folds =10.**

Name of the Dataset	Accuracy NB	Accuracy J48	Accuracy IBI
Diabetes			

Part-II

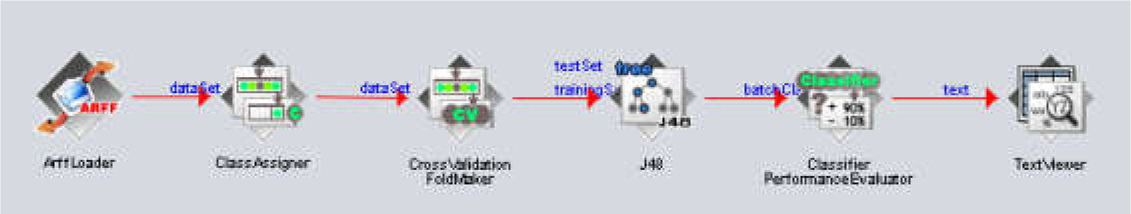
Working with WEKA Knowledge Flow Environment

Question 1: Develop a knowledge flow to find the classification accuracy for the dataset **My_First_Dataset.arff** using tree based **J48** classifier.

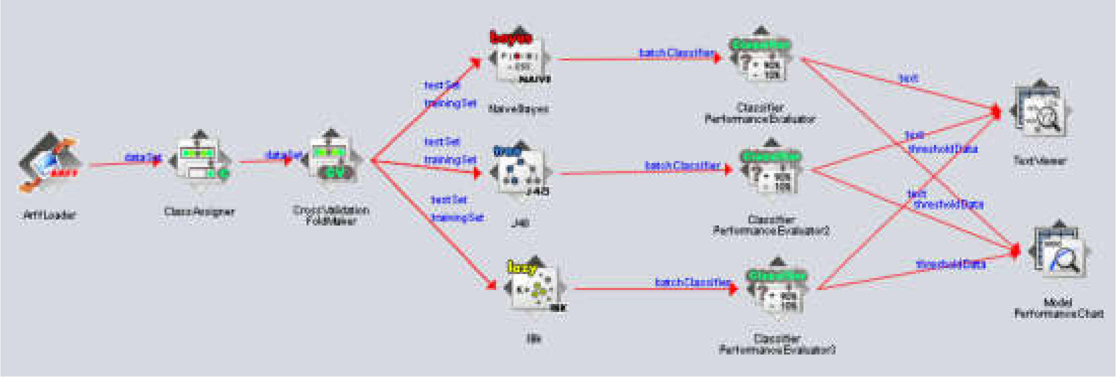
Steps:

- Go to Weka GUI chooser and select **Knowledge Flow**
- Go to Data Sources tab
  - Drag and drop the **Arff Loader** to Knowledge flow layout
  - Right click on the **Arff Loader** and choose the **configuration** and load the dataset **My_First_Dataset.arff**
- Go to Evaluation tab
  - Drag and drop the **Class Assigner** to Knowledge flow layout
  - Right click on the **Arff Loader** and choose the **dataset** and connect the **Arff Loader** and **Class Assigner** using rubber band connector
  - Right click on the **Class Assigner** and choose the **configuration** and choose the **class attribute**
- Go to Evaluation tab
  - Drag and drop the **Cross Validation Fold Maker** to Knowledge flow layout
  - Right click on the **Cross Validation Fold Maker** and choose the **configuration** to set the number of fold (default value is 10)
- Go to Classifier tab
  - Drag and drop **J48 classifier** to Knowledge flow layout

  - Right click on the **Class Assigner** and choose the **dataset** and connect with the **J48 classifier** using **rubber band connector**
  - Right click on the **Cross Validation Fold Maker** and choose the **training set** and connect with the **J48 classifier** using **rubber band connector**
  - Right click on the **Cross Validation Fold Maker** and choose the **test set** and connect with the **J48 classifier** using **rubber band connector**
- Go to the Evaluation tab
  - Drag and drop the **classifier performance evaluator** to Knowledge flow layout
  - Right click on the **J48 classifier** and choose the **batch classifier** and connect the **Classifier Performance Evaluator** using **rubber band connector**
- Go to the Visualization tab
  - Drag and drop the **Text viewer** to Knowledge flow layout
  - Right click on the **Classifier Performance Evaluator** and choose the **text** to connect the classifier performance evaluator using rubber band connector
- To start/run the work flow
  - (For Weka version upto 3.6.11) Right click on the **arff loader** and choose the **start loading**
  - (For Weka version 3.8.2) Click **Run this flow**
- To view the result
  - Do right click on the **Text viewer** and choose the **show results**
- To save the knowledge flow layout in the image format for publishing article or books
  - Press Control +Alt + shift + left click
  - Save in your desired image format



Question 2: Set a knowledge flow environment for computing the accuracy and ROC of the Naïve Bayes, tree based J48, and IB1 classifiers for the dataset diabetes with (number of fold 5) and identify which classifier gives better accuracy and ROC.



Part-II

Working with Java using the Weka API

Task 1: Write a Java Program to construct J48 classifier and display evaluation results for the dataset diabetes

```
import weka.classifiers.trees.J48;
 import
weka.core.converters.ConverterUtils.DataSource;
 import weka.classifiers.Evaluation;
 import java.util.Random;
 import weka.core.Instances;

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

DataSource source = new
DataSource("C:\\\\Program
Files\\\\WEKA_HOME\\\\data\\\\diabetes.arff");
 Instances data = source.getDataSet();
  if (data.classIndex() == -1)
   data.setClassIndex(data.numAttributes() - 1);
 String[] options = new String[1];
 options[0] = "-U";            // unpruned tree
 J48 tree = new J48();         // new instance of tree
 tree.setOptions(options);     // set the options
 tree.buildClassifier(data);   // build classifier
 Evaluation eval = new Evaluation(data);
 eval.crossValidateModel(tree, data, 10, new
Random(1));
System.out.println(eval.toSummaryString("\nResul
ts\n======\n", false));
    }
}
```

Question 1: Display only Correctly Classified Instances (Accuracy)

Accuracy of the J48 classifier

Task 2: Write a Java Program to select the significant features from the dataset diabetes using CFS and construct J48 classifier and display evaluation results

```
import weka.classifiers.trees.J48;
 import
weka.core.converters.ConverterUtils.DataSource;
 import weka.classifiers.Evaluation;
 import java.util.Random;
 import weka.core.Instances;
import weka.attributeSelection.AttributeSelection;
 import weka.attributeSelection.CfsSubsetEval;
import weka.attributeSelection.GreedyStepwise;
import
weka.classifiers.meta.AttributeSelectedClassifier;
public class WekaTest1 {
public static void main(String[] args) throws
Exception {
DataSource source = new
DataSource("C:\\\\Program
Files\\\\WEKA_HOME\\\\data\\\\diabetes.arff");
 Instances data = source.getDataSet();
 // setting class attribute if the data format does not
provide this information
 // For example, the XRFF format saves the class
attribute information as well
 if (data.classIndex() == -1)
   data.setClassIndex(data.numAttributes() - 1);
  AttributeSelectedClassifier classifier = new
AttributeSelectedClassifier();
  CfsSubsetEval eval = new CfsSubsetEval();
  GreedyStepwise search = new GreedyStepwise();
  search.setSearchBackwards(true);
  J48 base = new J48();
  classifier.setClassifier(base);
  classifier.setEvaluator(eval);
  classifier.setSearch(search);
  // 10-fold cross-validation
  Evaluation evaluation = new Evaluation(data);
  evaluation.crossValidateModel(classifier, data,
10, new Random(1));
System.out.println(evaluation.toSummaryString());

    }
}
```

Question 1: Display only Correctly Classified Instances (Accuracy)

Accuracy of the J48 classifier with CFS