COMP-4250 Big Data Analytics and Database Design

Project II
Data Mining with Weka

Deadline: End of Sunday March 28, 2021

Objective

The objective of this project is to gain hands-on experience using Weka (a popular data mining software) to build models from real world datasets. You will also evaluate different data mining algorithms in terms of accuracy and run time. This project is composed of five different tasks. Each task is explained in detail below. Note that **task 5 is optional** and you can get up to an extra 2% for performing it.

Software

The software to be used in this project is Weka. See the following link to download the latest version.
http://www.cs.waikato.ac.nz/ml/weka/
It is a free software that you can download and install on your machine. A guide on how to use Weka's Explorer can be found in the attached ExplorerGuide.pdf. Another document about the ARFF data format can be found in the attached Arff.pdf. Note that Weka is able to handle other data formats as well (such as csv). However, ARFF is the default format for Weka.

After installing Weka, you can use "java -Xmx256m -jar weka.jar" to modify the heap size when you invoke the program. You can increase the value of 256m if it is not enough. If not mentioned explicitly, you should use the default parameters of Weka for each classification algorithm.

Dataset: Along with this file on Blackboard.

Submission

- A pdf file that contains your answers to the tasks.
- The programs, if any, that you write for solving the optional task 5, and a readme file showing how to use these programs.
- Make a zip file from all of the above files, and name the file as LLL_FFF_DDD.zip, in which LLL, FFF, and DDD are your last name, first name, and student ID, respectively. One submission per team is enough, and you should name the team members at the top of your submitted pdf file.
- Submit the zip file on Blackboard before the deadline.

Task 1 (2%)

Consider the attached *lymphography* dataset (lymph.arff) that describes 148 patients with 19 attributes. The last attribute is the class attribute that classifies a patient in one of the four categories

(normal, metastases, malign_lymph, and fibrosis). Detailed information about the attributes is given in lymph_info.txt. The data set is in the ARFF format used by Weka.

Use the following learning methods (classification algorithms) that are provided in Weka to learn a classification model from the dataset with all the attributes:

- **C4.5** (weka.classifier.trees.J48)
- **RIPPER** (weka.classifier.rules.JRip)

For each learning method, report **only the classification model** learned from the dataset. Therefore, copy and paste the "Classifier model (full training set)" from Weka output to your report. For C4.5, it would be "J48 pruned tree". For RIPPER, it would be "JRIP rules:".

Task 2 (5%)

You are given a training dataset (monks-train.arff) and a test dataset (monks-test.arff) in which each training example is represented by seven **nominal** (categorical) attributes. The last attribute is the class attribute that classify each data point to one of the two classes (0 and 1). The attribute information is given below:

Attribute	Possible Values
A1	1, 2, 3
A2	1, 2, 3
A3	1, 2
A4	1, 2, 3
A5	1, 2, 3, 4
A6	1, 2
class	0, 1

Use the following learning methods provided in Weka to learn a classification model from the training dataset and test the model on the test dataset:

- **C4.5** (weka.classifier.trees.J48)
- **RIPPER** (weka.classifier.rules.JRip)
- **k-Nearest Neighbor** (weka.classifiers.lazy.IBk)
- **Naive Bayesian Classification** (weka.classifiers.bayes.NaiveBayes)
- **Neural Networks** (weka.classifiers.functions.MultilayerPerceptron)

Note that you have to use the "Supplied test set" option in the "Test options" box of Weka and pass the test data file (monks-test.arff) to Weka.

Report the classification summary, classification accuracy, and confusion matrix of each algorithm on test dataset. In other words, copy and paste the "Summary", "Detailed Accuracy By Class", and "Confusion Matrix" from Weka output to your report. Also, **briefly discuss your results in terms of accuracy**.

Task 3 (3%)

You are given a dataset on credit card application approval (credit.arff) in the ARFF format. The dataset describes 690 customers with 16 attributes. The last attribute is the class attribute describing whether the customer's application was approved or not. The dataset contains both symbolic and continuous attributes. Some of the continuous attributes contain missing values (which are marked by "?"). All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.
Randomly split the dataset into a training set (70%) and a test set (30%). This can be done using the "Percentage split" in the “Test option” box of Weka's "Classify" section (set the number to 70). Apply each of the following classification algorithms to learn a classification model from the training set and classify the examples in the test set.

- **C4.5** (weka.classifier.trees.J48)
- **Naive Bayesian Classification** (weka.classifiers.bayes.NaiveBayes)
- **Neural Networks** (weka.classifiers.functions.MultilayerPerceptron)

Report the classification accuracy of each learning algorithm on the test dataset. In other words, copy and paste the “Summary”, “Detailed Accuracy By Class”, and “Confusion Matrix” from Weka output to your report.

Note that C4.5, Naive Bayesian Classification, and Neural Networks can automatically handle both symbolic and continuous attributes as well as missing values of continuous attributes. Therefore, you do not need to do any extra preprocessing on the data and can directly run the above learning algorithms on the input dataset (credit.arff).

Task 4 (5%)

Conduct 10-fold cross validation to evaluate the following classification learning algorithms:

- **C4.5** (weka.classifiers.trees.J48)
- **RIPPER** (weka.classifier.rules.JRip)
- **Naive Bayesian Classification** (weka.classifiers.bayes.NaiveBayes)
- **k-Nearest Neighbor** (weka.classifiers.lazy.IBk)
- **Neural networks** (weka.classifiers.functions.MultilayerPerceptron)

on the following datasets from the UCI repository:

- Ecoli database (ecoli.arff) (https://archive.ics.uci.edu/ml/datasets/ecoli)
- Glass identification database (glass.arff) (https://archive.ics.uci.edu/ml/datasets/glass+identification)
- Image segmentation database (image.arff) (http://archive.ics.uci.edu/ml/datasets/image+segmentation)

Use all attributes to build the model. Report the classification accuracy and run time of each algorithm on each data set. Discuss the results and determine if there is an overall winner in terms of accuracy (misclassification rates) and run time.

You can summarize the results in two tables, one for the run time and the other for the accuracy. Then, you can add few sentences to discuss the results.

Task 5 (Optional and up to an extra 2%)

You are given a dataset (risk.csv) that describes 30,000 online purchase orders for an online trader. Each example in the dataset corresponds to an online purchase order and is described by 43 attributes. A detailed description of the attributes can be found in risk-attributes.txt. The second attribute is the target (i.e., class) attribute that indicates whether an order has a high risk of default payment. The class attribute has two values, "yes" meaning high risk and "no" meaning low risk. With the class attribute, the dataset contains 44 attributes in total. Randomly split the dataset into a training set (70%) and a test set (30%). This can be done using the "Percentage split" in the "Test option" box of Weka's "Classify" section (set the number to 70).

Your task is to help the online trader to recognize if a person who makes an order is a customer who will eventually pay the goods by using data mining techniques. You will use a classification algorithm to build a prediction model based on the training data. This prediction model shall then be used for classifying incoming orders into the high risk or low risk class. You need to assign each order in the test set to one of the two classes based on the prediction model learned from the training data.

For this learning and classification task, data preprocessing is very important. The dataset is a raw dataset and contains missing values and possibly irrelevant attributes. You may consider using feature selection (i.e., removing some attributes) and other data preprocessing techniques (filling NULL values, discretize values). For example, ORDER_ID will likely be an irrelevant attribute.

In addition to the classification result, you will also need to describe the data preprocessing methods (e.g., discretizing birth date into buckets) either from Weka or optionally written on your own and the classification method used in your solution (test different ones). If you write any programs for data processing, you should submit your programs with a readme file describing how to use the programs.

Report the **classification accuracy** (the same as previous tasks) and **misclassification cost** computed for the two classes that do not have equal weights. Below is the cost matrix for this dataset.

	High risk	Low risk
High risk	0	50
Low risk	5	0

Students who have higher classification accuracy and lower misclassification cost will get better mark for this task.

General Note:

When using data mining algorithms, you should be aware of the limitations of each algorithm. Some algorithms are not able to handle symbolic attributes or may perform poorly when data contains symbolic attributes. Some other algorithms may have this problem with continuous attributes. Also, some algorithms may not be able to handle missing values. In this case, missing values should be predicted in a preprocessing step before passing the data to the algorithm. Another option is to remove data points that contain missing values.