Here is a suggested methodology for incorporating WEKA into Chapter 5 of the text.

- The material in Sections 5.1 through 5.9 is not associated with a particular data mining tool. These sections can be covered without modification.

Section 5.10 Experimenting with WEKA using Attribute Filters

The WEKA toolbox offers several preprocessing algorithms to help determine a best set of attributes and instances for data mining. Here, we present three experiments. The first two experiments illustrate attribute selection. The third concentrates on instance selection.

Experiment 1: Attribute Selection - A simple approach

Attempting to build an efficient data model is an exercise in futility when presented with a wealth of irrelevant attributes. It is to our advantage to eliminate most or all irrelevant attributes prior to building a model. We have the option of developing our own attribute selection techniques or using one or more offered by the WEKA toolbox.

Let’s investigate a simple attribute selection method for supervised learning. Our experiment uses a dataset holding information about individuals who were either accepted or rejected when they applied for a credit card—see the *The Credit Card Screening Dataset* description box. The dataset contains 690 instances, 307 of which represent individuals who were approved to receive a credit card. The remaining 383 individuals had their credit card application rejected. We want to decide on a best set of attributes defining the classes contained in the data. Stated another way, we wish to test the possibility of building an accurate supervised learner model with a subset of attributes taken from the data.

The attributes and values have been mapped to a set of meaningless symbols to protect the confidentiality of the data. However, because the mapping is consistent, we should be able to apply data mining to analyze the dataset.
As a first step, open the WEKA explorer and load the dataset. Your screen will appear similar to the one given below:

Notice that we are in preprocess mode with *none* displayed in the filter command line. If we scroll the attribute window, we will see the last attribute designated as *class*. To perform our experiment, we first mine the data with J48 using all sixteen input attributes and *class* as the output attribute. The screen containing the resultant decision tree follows:
Notice that attribute *nine* represents the top-level node of the decision tree. Upon scrolling through the tree, we observe a classification accuracy of 84.7826%. Let’s see if we can do as well or better with fewer attributes. Using fewer attributes has several advantages two of which are building a simpler model in less time.

One straightforward approach to attribute selection uses a best-first strategy where at each iteration we build a decision tree, then record and remove the top-level attribute. For our example we first remove attribute *nine* and proceed to build a new decision tree. The second iteration shows *eleven* as the top-level node. Continuing, the next iteration shows *ten* as the next top-level node followed by *fifteen*. Let’s stop the process here.

Next, reload the original dataset, remove all attributes except for attributes *nine, ten, eleven, fifteen* and the *class* attribute. Next, we invoke J48 to build a decision tree. Our tree appears as below:
Notice that attribute *nine* is the only attribute making up the tree. The decision tree accuracy is 85.65%. This result is not significantly better than our original result using all attributes. However, we see that we are able to build a model as accurate as our original model using just one attribute. What conclusions might we reach from this result?

**Experiment 2: Attribute Selection using a WEKA filter.**

For the second experiment we use the same dataset but this time the attribute selection process takes place with the help of a WEKA preprocessing filter. Once again, load the Credit Card Screening Dataset. Our screen appears as follows:
To invoke one of WEKA’s attribute filters, we click on choose followed by filters- supervised-attribute-attribute selection. The resultant screen is given below:
Clearly, the command line showing the parameter list for the attribute selection filter is complex. We can see more filter options by simply clicking on the white space area on the command line. Doing so, we see the following:
For our experiment, we will use the default values shown. Close the above window and click on apply. The chosen attributes are given in the window below:

Notice that nine of the sixteen input attributes have been removed. Also, three of the four attributes initially chosen in the previous experiment are included.

Next, we invoke J48 to perform a data mining session. The outcome is displayed in the screen below:
Attribute *nine* is the top-level node. Four additional attributes make up the remainder of the tree. Scrolling, we see the cross-validation shows an accuracy of 84.9275%. We conclude that mining the data with a subset of the attribute set provides a model as accurate as the model developed using all sixteen original attributes.

**Experiment 3: Instance Selection**

WEKA supports several filters for instance selection. Instance selection filters are appropriate when we wish to use a smaller subset of our dataset. Some instance filters allow re-sampling thereby giving rare instances a more representative position within the data.

To illustrate instance selection, we apply instance re-sampling and PART to the *cardiology* dataset described in Chapter 2. This is the mixed form of the dataset containing both categorical and numeric data. Recall that the data contains 303 instances representing patients who have a heart condition (sick) as well as those who do not.

Load the cardiology-weka.arff file and bring the re-sample filter to the command line. We see the following screen:
Click in the *choose* white space area to bring up the following option screen.
Change the no-Replacement value to True and the sample size percent to 50.0. We invoke the PART rule generator and perform the data mining session.

This results in a classification accuracy exceeding 82%. Let’s further, limit the sample size to 15% and once again invoke PART. Our result displays a classification accuracy of more than 75%. This result lends support to the well-defined nature of this dataset.

Review Questions

1. Differentiate between the following terms:
   a. Data cleaning and data transformation
   b. Internal and external data smoothing
   c. Decimal scaling and Z-score normalization

2. In Section 5.4 you learned about three basic ways that data mining techniques deal with missing data while learning. Decide which technique is best for the following problems. Explain each choice.
   a. A model designed to accept or reject credit card applications.
   b. A model for determining who should receive a promotional flyer in the mail.
   c. A model designed to determine those individuals likely to develop colon cancer.
   d. A model to decide whether to drill for oil in a certain region.
   e. A model for approving or rejecting candidates applying to refinance their home.
Data Mining Questions

1. Repeat Experiment 1 described in Section 5.10 using the *deerhunter* dataset.
   Give a detailed report of your results.

2. Perform Experiment 2 from Section 5.10 using the *cardiology-weka.arff* dataset.
   Repeat the experiment but this time apply an alternative attribute filter. Compare
   the result of the two experiments.

3. The *cardiologyNN-weka.arff* data file contains the same instances as the
   *Cardiology-weka.arff* file but the categorical attributes have been changed to
   numeric equivalents.
   a. Load the *cardiologyNN-weka.arff* file into the Weka explorer. Please refer to
      Table 2.1 to see how the categorical attributes are mapped to corresponding
      numerical equivalents. For example, the table shows that values male and
      female for attribute sex are replaced with a 1 and a 0. Likewise the values
      angina, abnormal angina, noTang, and asymptomatic for attribute chest pain
      type are respectively replaced with 1,2,3, and 4. Note that the class attribute
      represents an instance of the healthy class with a 1 and an instance of the sick
      class with a 0.
   b. Perform the first experiment in Section 5.10 using this dataset.

4. The Weka samples directory includes a file named *Grb-Weka.arff*. Read about
   this dataset in the description box titled *The Gamma-Ray Burst Dataset* located at
   the end of this chapter. This exercise requires you to use SimpleKMeans to
   perform four unsupervised clusterings of this data. For the first experiment,
   indicate that you desire two clusters. Be sure to eliminate the attribute giving the
   burst number. Use p 256, hr32 and T50. Eliminate f1, hr321 and T90. Record the
   mean and standard deviation values for each cluster.

   For the second experiment, use the same attributes but indicate that three clusters
   are to be formed. Once again, record the results.

   For the third experiment use f1, hr321 and T90 as the only input attributes and
   indicate that two clusters are to be formed.

   For the fourth experiment, repeat the third experiment but indicate that three
   clusters are desired.
   a. For each of the four clusterings, examine the cluster means and standard
      deviations. Provide a brief written description of the nature of the bursts
      falling into each cluster. For each of the four clustering experiments, indicate
      any marked differences you find in average burst length, burst brightness, and
      burst hardness.
b. Make a decision about which of the four clusterings is a best representation of the data. Justify your answer.

5. Visit the FTP site: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/ (or any of the Web sites listed in Appendix B) and select a dataset that interests you. Download and format the data as an .arff file. Perform one or more data mining experiments with the data.

**Computational Questions**

1. Set up a general formula for a Min-Max normalization as it would be applied to the attribute age for the data in Table 2.3. Transform the data so the new minimum value is 0 and the new maximum value is 1. Apply the formula to determine a transformed value for age = 35.

2. The price of a certain stock increases from $25.00 to $40.00. Compute the percent increase in the stock price.

3. You are to apply a base 2 logarithmic normalization to a certain numeric attribute whose current range of values falls between 2300 and 10,000. What will be the new range of values for the attribute once the normalization has been completed?

4. Apply a base 10 logarithmic normalization to the values for attribute age in Table 2.3. Use a table to list the original values as well as the transformed values.

5. Use the CreditCardPromotion.xls data file together with the initial element population shown in Table 5.1 to perform the first iteration of the genetic algorithm for attribute selection. Use 10 instances for training and the remaining instances as a test set. Use classification correctness on the test data as your fitness function. List the fitness scores for each of the three elements of the initial population.

6. Based on the experiments in this chapter, can you hypothesize an inverse relationship between a setting for the real-tolerance parameter and attribute significance?