Assignment 2: Modeling on WEKA

Due: Thursday, 20 Sep 2018

In this assignment you will explore and experiment with C.45, Naïve Bayes, and K-Nearest Neighboor classification algorithms using WEKA. The dataset for this assignment can be downloaded from the Elisa.

1. In this problem we will use the PEP data (**bank-data.zip**) for the purpose of target marketing. In this case, we plan on using the historical data from past customer responses (the training data from last assignment) in order to build a classification model. The model will then be applied to a new set of prospects to whom we may want extend an offer for a PEP. Rather than doing a mass marketing campaign to all new prospects, we would like to target those that are likely to respond positively to our offer (according to our classification model).

   There are two data sets available (in ARFF format) contained in the Zip archive **bank-data.zip:**

   - **bank-data.arff**: Pre-classified training data Set for Building a Model
   - **bank-new.arff**: A set of new customers from which to find the "hot prospects" for the next target marketing campaign (i.e. those that are likely to respond positively to an offer for PEP.

   Note that since the ID attribute is not used for building the classifier, you should begin by loading each of these data sets into WEKA, and in each case removing the ID attribute and saving both filtered data sets into new files.

   a. Using **WEKA** package create a "C4.5" classification model based on the pre-classified training data. In WEKA, the C4.5 algorithm is implemented by "**`weka.classifiers.trees.J48`**". Use 10-fold cross-validation to evaluate your model accuracy. Record the final decision tree and model accuracy statistics obtained from your model. Be sure to indicate the parameters you use in building your classification model (if you experiment with non-default values). You can save the statistics and results by right-clicking the last result set in the "Result list" window and selecting "Save result buffer." You should also generate and create a screen shot of your tree by selecting the "Visualize tree" command from the same menu **[Note: you can resize the window as necessary, right-click inside the window, and select the command "Fit to Screen" to get a better view of the full tree]**. You should provide the decision tree together with the accuracy results from the cross-validation as part of your submission.
   b. Next, apply the classification model from the previous part to the new customers data set as the "Supplied test set." Be sure to the select the option "**Output predictions**" in the **test options** for the classifier (under **More Options**). This option will show you the predicted classes for the 200 new instances. In your final submitted result you shouldo map the resulting answers back to the original customer "id" field for the new customers (this could be done using a spreadsheet program such as Excel and the original new customers data set in CSV format). **Provide your resulting predictions for the 200 new cases and other supporting documentation as part of your submission**.

2. In this problem you will use Naïve Bayesian Classification on usage data associated with a hypothetical ecommerce Web site to determine if a user will return to the site in the future. The data set (**Visit-Nominal.csv**) contains a set of 100 user sessions involving activities on the Web site. The attributes in this data set have been converted into categorical (nominal) binary attributes indicating whether the user has visited a specific section of the site or has purchased a product in the past visits. The attributes are described as follows:

- **Home** - indicating whether the user has visited the homepage.
- **Browsed** - indicating whether the user has spent time (using some pre-specified threshold) browsing the product catalog.
- **Searched** - indicating whether the user has performed searches for specific products.
- **Prod_A, Prod_B, Prod_C** - indicating whether the user has purchased products belonging the corresponding product category.
- **Visit_Again** - the class attribute indicating whether the user has subsequently returned to the site in a future session.

Your tasks in this problem are as follows:

a. Load the data set into WEKA and under the Classify tab choose **classifiers.bayes.NaiveBayesSimple**. Under the **Test options** select **Use training set**. Then run the classifier and save the result set buffer. You will notice that the model specified the conditional probabilities associated with different attributes for each of the two classes (**Visit_Again=yes** and **Visit_Again=no**). For example, using this information you can find **Pr(Browsed=no | Visit_Again=yes)** or **Pr(searched=yes | Visit_Again=no)**. Also, the model includes the prior probabilities of each of the two classes, **Pr(Visit_Again=no)** and **Pr(Visit_Again=yes)**. Submit your result set as part of your answer.

b. Next, using the probabilities you obtained from the model and Bayes' Rule, manually compute the probabilities of each of the following two new instances belonging each of the two classes:
   i. New instance **X = <Home=yes, Browsed=no, Searched=yes, Prod_A=no, Prod_B=yes, Prod_C=no>**
   ii. New instance **Y = <Home=yes, Browsed=yes, Searched=no, Prod_A=yes, Prod_B=no, Prod_C=yes>**

   For example, in the case of X, you must use Bayes' rule to compute **Pr(X | Visit_Again=yes) and Pr(X | Visit_Again=no)**, and similarly for Y. Show the details of your computation.

3. For this problem you will use an **segment.zip** and perform classification based on the **K-Nearest-Neighbor (KNN)** Approach. This dataset contains information characterizing images with each line corresponding to one image. Each image is represented by 19 features (these are the columns in the data and correspond to the feature names in the list of attributes. The last column in the data contains the class labels corresponding to the image types (brickface, sky, foliage, cement, window, path, grass). The data set contains three files. The file "**segment-train.arff**" is the training data consisting of 30 instances (images) from each of the 7 categories. The test data ("**segment-test.arff**") is used for evaluating the model built using the training data and it contains 2310 instances. A detailed description of the data set, including the meanings of various attributes is provided in the file "**segment-decription.txt**".

   Your tasks in this problem are the following:

   a. Load in the training image segment data into WEKA and select WEKA's KNN implementation under the Classify tab. This implementation is called IBk and it is located in the module: **weka.classifiers.lazy.IBk**. Open the classifier options dialog box and select an appropriate value for K (number of neighbors). Under **Test options** choose 10-fold cross-validation (which is the default). Run the classifier multiple times, experimenting with different values of K (you may wish to try 5, 10, 15, 20 and so on) and with or without the distance weighting option set. For each run examine the evaluation result. Once you are satisfied that you have the best set of options, record the final results by saving your buffer for the corresponding result set. You should submit this result set and also provide a 1-2 paragraph summary of which options you tried and your findings.

b. Next, apply your model from part (a) to the test data. Under the **Test options** select "**Supplied test set**" and set the test set to the file **segment-test.arff**. Under More options, make sure that "Output predictions" is selected. Finally, run the KNN classifier on the test data. Compare the evaluation results to the results from 10-fold cross-validation. Submit your results set (including the predictions) along with a summary of your observations.