SENG 474, CSC 503: Data Mining: Fall 2019 – Assignment 1.

1. (4 points) Construct the root and the first level of a decision tree for the titanic dataset. Use the ID3 algorithm. Show the details of your construction (entropies calculated for each step). You can use a spreadsheet or SQL database to compute the counts.
Then, check your solution with Weka and submit a text file of your classifier output window.

2. (4 points) Construct two rules using PRISM for the weather dataset. Show the details of your construction. Then, check your solution with Weka and submit a text file of your classifier output window.

3. (4 points) Classify using Naïve Bayes method on the titanic dataset the data items:

```
2nd     child   male    ?
2nd     adult   female  ?
```

Then, check your solution with Weka (the dataset is included with Weka).
Note: You can download Weka from: http://www.cs.waikato.ac.nz/ml/weka

4. (10 points) Implement a multinomial Naive Bayes classifier for text classification. This classifier will be used to classify fortune cookie messages into two classes: messages that predict what will happen in the future and messages that just contain a wise saying. We will label messages that predict what will happen in the future as class 1 and messages that contain a wise saying as class 0. For example,

- "Never go in against a Sicilian when death is on the line" would be a message in class 0.
- "You will get an A in SENG 474" would be a message in class 1.

You can use any language you wish. There are two sets of data files provided:

1. The training data:
    - **traindata.txt**: This is the training data consisting of fortune cookie messages.
    - **trainlabels.txt**: This file contains the class labels for the training data.
2. The testing data:
    - **testdata.txt**: This is the testing data consisting of fortune cookie messages.
    - **testlabels.txt**: This file contains the class labels for the testing data. These are only used to determine the accuracy of the classifier.

Your results must be stored in a file called results.txt.

1. Run your classifier by training on traindata.txt and trainlabels.txt then testing on traindata.txt and trainlabels.txt. Report the accuracy in results.txt (along with a comment saying what files you used for the training and testing data). In this situation, you are training and testing on the same data. This is a sanity check: your accuracy should be very high i.e. > 90%
2. Run your classifier by training on traindata.txt and trainlabels.txt then testing on testdata.txt and testlabels.txt. Report the accuracy in results.txt (along with a comment saying what files you used for the training and testing data). We will not be letting you know beforehand what your performance on the test set should be.

Submit your source code and the results.txt file.
Reference: This exercise is adapted from Weng-Keen Wong, Oregon State University.