COMP3308 Artificial Intelligence
Semester 1, 2014

Assignment 1. Predicting Diabetes

Submission deadline: Friday, week 8, 5pm (2 May).
Late policy submission: A penalty of -1 mark will apply for each day late and the assignment will not be accepted if it is submitted more than 7 days after the due date. The cut-off time is 5pm.
This assignment can be completed individually or in pairs. Working in pairs is encouraged. Both students will receive the same mark.
Submission instructions:

- Submit your assignment electronically via eLearning. No printed version is required.
- Your submission should include: report, code, the three data files (`pima.csv, pima-CFS.csv` and `pima-folds.csv`) and an assignment cover sheet (group or individual, signed and scanned; it is available from: http://sydney.edu.au/engineering/it/current_students/postgrad_coursework/policies/academic_honesty.shtml)
- All files should be zipped together in a single file. The zip file should be named 0123456.zip, where 0123456 is your SID. In case of a pair submission, put both SIDs separated by an underscore: 0123456_0789123.zip. Only one of the two students needs to submit.

Programming language: You can write the program in any of the following languages: Python, Java, C, C++ or Matlab. We will need to be able to test your code on the University machines, so you must ensure that your code is compatible with the version of the language available on those machines, and include instructions on how to run your code. Note that while some of these languages have built-in classification libraries, you are not permitted to use them for the purposes of this assignment, and must implement your own.
Weight: The assignment will be marked out of 20, and will form 20% of your final mark.

The goal of this assignment is to:

1) Implement the k-Nearest Neighbor and Naïve Bayes algorithms, and also the stratified cross validation method,
2) Evaluate the classification performance of the two implemented algorithms and other classifiers from Weka on a real dataset,
3) Investigate the effect of feature selection, in particular the Correlation-based Feature Selection method (CFS) from Weka.

1. Data
The dataset for this assignment is the Pima Indian Diabetes data. It can be downloaded from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/. It contains 768 instances described with 8 numeric attributes. There are 2 classes (0 and 1). Each instance corresponds to a patient's record; the attributes are personal characteristics and test measurements; the class shows if the person shows signs of diabetes or not. The patients are from Pima Indian heritage, hence the name of the data.

Download the data from the repository. There are 2 files associated with each dataset: `*.names` describing the data (e.g. number and type of attributes and classes, and their meaning) and `*.data` containing the data. **Your task is to predict the class - class 0 or class 1.**

2. Data preprocessing

- Read the `pima-indians-diabetes.names` file and learn more about the meaning of the attributes and the classes.
- The `pima-indians-diabetes.data` file is in CSV format. Add a header line for the features based on the information from the names file, e.g. "num_pregnant" for feature 1, etc. Change the class value from numeric to nominal, e.g. 0 →"class0", 1 →"class1".
- There are missing attribute values in this dataset, which have unfortunately been coded as zeroes (see note on the dataset description page). The result is that some instances have biologically

impossible values for certain attributes, adding significant noise to the dataset. As the dataset description states, *"use your best judgement and state your assumptions"*: you must work out which values are missing, decide on an appropriate solution, and describe your approach and the reasoning behind it in your report. You might find it useful to load the dataset into Weka and look at the per-attribute histograms in the "Preprocess" tab.

- Normalise the values of each attribute to get values in the range [0,1]. You can do this in Weka; it has an in-build normalisation filter for this. The normalisation should be along each column (attribute) not each row (instance). The class attribute is not normalised — it should remain unchanged.
- Save the preprocessed file as `pima.csv`.

3. Implement the two classification algorithms:

- k-Nearest Neighbor with Euclidean distance.
- Naïve Bayes. As the features are numeric, you will need to implement the version for numeric attributes using a probability density function. Assume a normal distribution, i.e. use the probability density function for a normal distribution.

Your program should be able to read the CSV file.

4. Implement 10-fold stratified cross validation for evaluating the performance of two classifiers. For each 10-fold cross validation run, your program should print the accuracy of the classifier on the test set (the fold not used for training), and at the end the average accuracy over the 10 runs.

We also would like to test if you have implemented the 10-fold stratified cross-validation correctly. Print the 10 stratified folds to a file called `pima-folds.csv` in the following format:

```
fold1
Examples from this fold in CSV format, one per line, e.g.
0.588,0.628,0.574,0.263,0.136,0.463,0.054,0.333,class1
<empty line>
fold2
examples from this fold
<empty line>
…
fold10
examples from this fold
```

Note that the number of instances per fold should not vary by more than one, i.e. if the total number of instances is not divisible by ten, you should distribute the remaining items amongst the folds rather than placing them all into one fold.

5. Apply feature selection using CFS

CFS [1] is a method for selecting a subset of the original attributes. It searches for the 'best' subset of features where 'best' is defined by a heuristic which takes into consideration two criteria: 1) how good the individual features are at predicting the class and 2) how much they correlate with the other features. Good subsets of features contain features that are highly correlated with the class and uncorrelated with each other.

[1] Hall, M.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. 17th Int. Conf. on Machine Learning (ICML). Morgan Kaufmann (2000) 359-366.
http://waikato.researchgateway.ac.nz/handle/10289/1024

Load the `pima.csv` file in Weka, and apply CFS to reduce the number of features. It is available from the "Select attributes" tab in Weka. Use "Best-First Search" as the search method. Save the CSV file with the reduced number of attributes (this can be done in Weka) and name it `pima-CFS.csv.`

6. In WEKA select 10-fold cross validation (it is actually 10-fold *stratified* cross validation) and run the following algorithms: ZeroR, 1R, k-Nearest Neighbor (k-NN; IBk in Weka), Naïve Bayes (NB), Decision

Tree (DT; J48 in Weka) and Multi-Layer Perceptron (MLP). Compare their performance with your k-Nearest Neighbor and Naïve Bayes classifiers. Do this for the case without feature selection (using `pima.csv`) and with CFS feature selection (using `pima-CFS.csv`).

7. Write a report (similar to a research paper) describing your analysis and findings. It should include the following sections:

1) Aim – briefly state the aim of your study (e.g. predicting X based on Y etc.) and write a paragraph about why the problem is important.

2) Data
   - Data set used – Briefly describe the dataset. Mention the number of attributes and classes.
   - Data preparation – Provide a summary of the preprocessing applied.
   - Attribute selection – Briefly describe the CFS method. List the attributes selected by CFS.

3) Results and discussion
   - Results – Present the accuracy results (in %, using 10-fold cross validation) in the following table where My1-NN, My5-NN and MyNB are your implementations of the 1-NN, 5-NN and NB algorithms, using your stratified 10-fold cross validation.

Accuracy on test set [%]

	ZeroR	1R	1-NN	5-NN	NB	DT	MLP
No feature selection							
Correlation-based feature selection							

	My1-NN	My5-NN	MyNB
No feature selection			
Correlation-based feature selection			

   - Discussion – Compare the performance of the classifiers, with and without feature selection. Compare your implementations of k-NN and NB with Weka's. Discuss the effect of the feature selection – did CFS select a subset of the original features, and if so, did the selected subset make intuitive sense to you? Was feature selection beneficial, i.e. did it improve accuracy, or have any other advantages? Why do you think this is the case? Include anything else that you consider important.

4) Conclusions – Summarise your main findings and, if possible, suggest future work.

5) Reflection – What was the most important thing you learned from this assignment? [1-2 paragraphs]

6) Instructions on how to run your code.

COMP3308 Assignment 1 – Marking Sheet
Marked out of 20

Student(s):

	Your mark	Comments
1. [9 marks] Report [0.5 marks] Introduction - What is the aim of the study? - Why is this study (the problem) important? [1.5 marks] Data – well explained - Dataset - Data preparation – preprocessing steps applied - Attribute selection – brief summary of CFS and stating the selected attributes [4 marks] Results and discussion - All results presented - Correct and deep discussion of the results - Effect of the feature selection – beneficial or not (accuracy, training time, other advantages) - Comparison between the classifiers (accuracy, training time, other advantages) [1.5 marks] Conclusions and future work - Meaningful conclusions based on the results - Meaningful future work suggested [0.5 marks] Reflection (meaningful and relevant personal reflection) [1 marks] English and presentation - Academic style, grammatical sentences, no spelling mistakes - Good structure and layout; consistent formatting		
2. [1 mark] Data preparation is correct - Headers added to the CSV file - Normalization correctly applied for each attribute (each data column except the class) - CFS correctly applied		
3. [9 marks] Code - Stratified cross validation is correct - k-NN correctly implemented and prints the results as required - NB correctly implemented and prints the results as required		
4. [1 mark] At the discretion of the marker - for impressing the marker with something		
Penalties: -2 mark maximum for badly written code or code that is not well documented and difficult to read -0.5 marks for not including instructions on how to run your code		
Penalty for late submission: -1 mark for each day late		
Total (out of 20):		