University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2018

Assignment 2: Classification

Due: Wednesday, March 21, 2018 (by 11:59pm)

Assignment Type*:* group, the max team members is 3.

Hand in**:** one report (PDF) per group, via the LEARN dropbox. Also submit the code / scripts needed to reproduce your work.

Objective**:** To gain experience on the use of classification methods. The emphasis is on analysis and presentation of results not on code implemented or used. You can use libraries available in MATLAB, python, R or any other programs available to you. Some instructions for MATLAB are provided, you need to mention explicitly the source of any other references used.

Data sets (available on the UW 'LEARN' system)

Dataset A (bank-additional-full.csv) Online at : https://archive.ics.uci.edu/ml/datasets/bank+marketing	The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. There are four datasets. Use dataset bank-additional.csv for this assignment. bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).
Dataset B (DataDNA.mat)	This data is the splice junctions on DNA sequences. The given dataset includes 2200 samples with 57 features, in the matrix 'fea'. It is a binary class problem. The class labels are either +1 or -1, given in the vector 'gnd'. Parameter selection and classification tasks are conducted on this dataset.

I. [40] Revisiting HW5 Bank Classification with New Tools (for dataset A)

1) [5] Load a simple dataset and perform some basic data preprocessing to fill out "unknowns", outliers or other invalid data. Explain what preprocessing was performed and why.
2) [5] Divide data into train and test portions, justify your split decision.
3) [10] Apply classification using Decision Trees (DT), Random Forests (RF) and Neural Networks (NN) and run using standard libraries in your language or choice. Indicate the classification properties (example: depth of tree, size of neural network, ensembles for RF) you have chosen and justify. Briefly describe the algorithms employed by the libraries you are using (example ID3, C4.5, C5.0 and CART for DT). Make sure to run the classification on 10 features or more.

4) [10] Create a few plots of your model on the test data, two of the data dimensions at a time, indicating the predicted elements of each class using different colors or shapes. You may need to try plotting various pairs of dimensions to see which provide some interesting result. Be sure to label your axis and legend. Why is separation better on some plots than others?

5) [5] Produce a table with the true/false positive/negative metrics as well as accuracies. Compare the values using bar charts. HINT: **classification_report from sklearn.metrics**

6) [5] Provide a short explanation of the results you have shown and what it means. Which classification method performed better? Why? Contrast performance with classification from the previous homework and comment on the difference, if any.

7) **[+5] For Fun/Bonus:** attempt at least one method to tackle the discrepancy in the size of the classes (imbalanced data).

II. [60] Parameter Selection and Classification (for dataset B)

Classify dataset B using four classifiers: k-NN, Support Vector Machine (with RBF kernel), Random Forests and simple Neural Networks (MLPs). The objective is to experiment with parameter selection in training classifiers and to compare the performance of these well- known classification methods.

1) [5] Preprocess the given data using the Z-score normalization on the data. Justify the choice of Z-score normalization here, as opposed to min-max normalization. Why do you need normalization in general? Justify why you would normally split the test and training set randomly. What is the distribution of the +1,-1 classes in the dataset?

2) Parameter Selection:

   a) [7.5] For k-NN you need to evaluate the best value **k** to use. Using 5-fold cross validation (the `crossvalind` function can help) on the training set evaluate k-NN on the values k=[1, 3, 5, 7, …, 31]. Plot a figure that shows the relationship between the accuracy and the parameter **k**. Report the best **k** in terms of classification accuracy. Explain why you didn’t evaluate directly on the test set.

b) [7.5] For the RBF kernel SVM, there are two parameters to be decided: the soft margin penalty term "**c**" and the kernel width parameter "**sigma**". Again use 5-fold cross validation on the training set to select the parameter "**c**" from the set [0.1, 0.5, 1, 2, 5, 10, 20, 50] and select the parameter "**sigma**" from the set [0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10]. Report the best parameters in terms of classification accuracy including plotting the ROC curves.

3) Train (at least) six classifiers and report the results:

a) [15] Classify the test set using k-NN, SVM, Random Forests and Neural Networks. Use the chosen parameters from the parameter selection process in question 2 for k-NN and SVM. For the next two classifiers use the default setups listed at the end for Random Forests and Neural Networks.

b) [10] For the fifth and sixth classifiers, you should explore the parameters of the Random Forests and Neural Network models to devise your own classifier instance that does better than the other methods. For example, you could consider a deeper neural network with multiple layers, use different optimization/solver algorithms, you could modify the Random Forests using different parameter settings for depth and number of trees or enable boosting. Play around with options and choose a setting for RFs and NNs that performs better.

c) [5] Repeat each classification method 20 times by varying the split of the training-test set as in II.2. Report the average and standard deviation of classification performance on the test set regarding: accuracy, precision, recall, and F- Measure. Also report the training time and classification time of all the methods. Explain why the classification was repeated 20 times.

4) [5] Comment on the obtained results, what are the benefits and weaknesses of each method on this dataset. How could this analysis help to make the choice of the right method to use for a dataset of this type in the future?

5) [5] If you had to remove 1 features from the dataset, which feature would you select to remove from the dataset and why? What would have happened if you did classification on two dimensions only?

Notes:

1. For classification methods of k-NN, Naive Bayes, SVM, and Decision trees and Random Forests (bagged decision trees), Matlab has implemented functions as: `knnclassify`, `svmtrain/svmclassify,` and `fitctree /eval, and TreeBagger`

   Python has implemented functions in Sklearn:
   http://scikit-learn.org/stable/modules/neural_networks_supervised.html

   http://scikit-learn.org/stable/modules/tree.html

   http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

   http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

   http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

   `For simple Neural Networks`

2. There is a well-known SVM library named libSVM, which is implemented in C++ and includes a matlab interface (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`)

3. For the clustering methods of k-means, fuzzy c-means, and hierarchical clustering, Matlab has implementation as `kmeans`, `fcm`, and `linkage/clusterdata`.

4. For Neural Networks in Matlab you should be able to use the Matlab `patternnet` function. See (`http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html`) for a tutorial on setting up neural networks visually and generating the initial code. For an example of a deeper network built out of autoencoders look at (https://www.mathworks.com/help/nnet/examples/training-a-deep-neural-network-for-digit-classification.html). For python the sklearn package MLPClassifer provides similar properties: http://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification

5. Late submissions (up to 3 days) are accepted with penalty of 10% per day.

Default Parameters:
For easier comparison of results please try to use default parameters for the first four classifiers of part II. For all other parts you can select your own parameters but you must justify your choices.

Decision Trees:

- If using Python or R you can find the default values used in matlab at `https://www.mathworks.com/help/stats/fitctree.html`

Property	Value
Algorithm	Standard CART
Depth	n - 1 for MaxNumSplits. n is the training sample size. 1 for MinLeafSize. 10 for MinParentSize.
MaxNumCategories	10
Pruning	on

Random Forests:

- If using Python or R you can find the default values used in matlab at `https://www.mathworks.com/help/stats/treebagger.html`

Property	Value
'MinLeafSize'	1
Tree properties same as above	

Neural Networks:

- If using Python (MLPClassifier) or R you can find the default values used in matlab at `http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html`

Property	Value
hiddenLayerSizes Row vector of one or more hidden layer sizes	10 (single layer) enter an array or sizes to create multiple fully connected layers
Activation	Tanh
Training Algorithim	Scaled conjugate gradient backpropagation set solver='sgd'