CS 6375- ASSIGNMENT 2

Please read the instructions below before starting the assignment.

- Please submit the deliverables for each step in a different folder labeled Step1, Step2, etc.
- In the code folder, please include a README file indicating how to compile and run your code. Also, mention clearly which packages you have used.
- You should use a cover sheet, which can be downloaded at: http://www.utdallas.edu/~axn112530/cs4375/CS4375_CoverPage.docx
- You are allowed to work in pairs i.e. a group of two students is allowed. Please write the names of the group members on the cover page.
- The deadline for this assignment is Monday October 17, 2016. No extensions are allowed.
- You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After that, there will be a penalty of 10% for each late day. **The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions through Piazza, and not through email.
- In many scenarios, you might have to use your best judgement. We want to see that you have put in a sincere effort in all steps of this assignment and have evaluated the classifiers fairly.

ASSIGNMENT 2

In this assignment you will compare the performance of the classifiers that you have learned so far in the course.

The steps of this assignment are outlined below:

I. Choose dataset:

UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html) is one of the most popular sources of datasets. You can filter them for classification tasks and also choose other characteristics such as attribute type and application areas e.g. Life Sciences, CS/Engineering, etc.

For this assignment, you need to choose a dataset that interests you and that satisfies the following criteria:

1. The task should be "Classification", since you would be implementing classifiers.
2. The data type should be "Multivariate"
3. The number of instances should be more than 100.

It would be advisable to check how many missing or N/A values your dataset contains. Please note that image and text data can be tricky to analyze and requires special pre-processing. If you choose such a dataset, you are responsible for performing pre-processing before using classifiers.

Deliverables for this step:
Summary of your chosen dataset, containing the following information:
- number of instances and attributes
- number of classes in the predicted variable
- if there are a number of NA or null values, how you plan to handle that situation
- Any other pre-processing that may be needed.

II. Understand the dataset:

On the UCI website, corresponding to each dataset there is a list of “Relevant Papers”. They describe how this dataset was used, and the results of the authors’ experiments. The results could be in the form of an accuracy metric or any other form of evaluation metric.

Deliverables for this step:

For the second part of the assignment, please read the relevant papers and write a summary that contains the following details:

- How was the data obtained (source of the data)
- What did the authors use the dataset for (use cases for the dataset)
- What type of experiments were done on the dataset (experiments)
- Summary of the authors' results (this could be the accuracy that the authors obtained or other forms of results).
- Any other information you found interesting

It is up to you how many papers you read. The idea is to understand what the data was used i.e. what type of analysis. You can get this by reading just one paper or you might have to read several. Please use your judgement.

III. Pre-process the dataset:

Pre-processing involves checking the attributes and predicted values for null values, scaling the datasets, checking for redundant attributes (those that have no correlation with the output or those that are all same), and finding out correlations between attributes themselves and the output values.

You will need to perform the pre-processing activities for your chosen dataset. Remember that some of the classifiers, like neural nets, work well with scaled data. So, it's a good idea to pre-process and scale the dataset before using it in the next steps.

Deliverables for this step:

- Brief summary of your pre-processing strategy
- Code that you used for pre-processing
- It would also be good to include plots for correlation between the various variables and a histogram the variables showing data distribution. This idea was explained in the lab sessions.

IV. Implement classifiers on the dataset:

In this section, you will implement the five classifiers that you have learned so far on your chosen dataset. You will need to divide the dataset into training and test parts, and train your classifier using the training subset and test it on the testing subset. Further details will be explained below.

A. Classifiers to train:

So far, we have studied the following classifiers and you will train them on your chosen dataset:

- Decision Trees
- Perceptron (Single Linear Classifier)
- Neural Net
- Support Vector Machines
- Naïve Bayes Classifiers

You are free to use any packages or libraries for these classifiers. I would strongly encourage you to use R, since we covered some of the basics in lab sessions in class.

The suggested packages for the various classifiers are:

Classifier	**Suggested packages in R**
Decision Trees	rpart
Perceptron	neuralnet (Remember to set “hidden” parameter to 0)
Neural Net	neuralnet
SVM	e1071
naïve Bayes	e1071

You are free to use other R packages, just be sure to mention them in the README file.
Also, in order for your code to exit gracefully in case of package error, please be sure to include require statements at the top of your code, something like:
> require(packagename)

B. Classifier parameters

Each classifier requires a number of parameters. For example, a neuralnet model requires you to specify number of hidden layers, error threshold, number of different repetitions, etc. Similarly, a SVM model requires you to specify the kernel and cost parameters. In order to get the best performance from each classifier, **<u>you have to train the classifier using different set of parameters</u>**. This is a critical part of the assignment that is overlooked by many students.

In order for us to see that you really did try a number of different parameters, **<u>you have to maintain a log of your experiments.</u>**
A tabular form as shown below would be great:

Experiment #	**Classifier**	**Train/Test Ratio**	**Parameter1**	**Parameter2**	**…**	**Outcome (Accuracy)**
1	NeuralNet	80/20	hidden = 2	error = 0.5	…	86%
2	NeuralNet	80/20	hidden = c(5, 3)	error = 0.1		91%
…	…		..	..	.	.

It is up to you how many times you run the experiments or how you many permutations of the parameters you try. However, you need to convince us that you made a sincere effort for finding the best set of parameters.

C. Experimental Methodology

You will need to split the dataset **randomly** into training and test partitions. For this assignment, the suggested ratio of training and test partitions is 80:20 (you can change this to 90:10 or 95:5, but be sure to mention it in the README file). R has a function called **sample** that you will find useful. For each of the classifiers, you will build a model using the training partition and check its performance on the test partition and report testing accuracy.

To get a better idea of accuracy, you will have to run your experiments 5 times, each time choosing a different random sample for training and testing partitions. The pseudocode for the experimental methodology is explained below:

```
C = List of classifiers

for numSamples in 1:5
    // create training data by sampling 80% of the dataset
    training = sample(d, 80)
    // create test data by removing train data from the dataset
    test = d – {training}
    for c in classifiers C
        // create a model of type c using train
        model <- createModel(c, train)
        // find accuracy of model of type c on test
        accuracy <- findAccuracy(model, test)
        write accuracy to file
    next c
next numSamples
```

D. Reporting your results:

After finding the best parameters for each of the classifiers in step B, you would run the classifiers (with the best parameters) using the approach and algorithm presented in step C.

This will allow you to compare the performance of each of the classifiers on the dataset. Output your results in a table format. A sample is shown below.

Sampling#	Train/Test Percent split	Decision Tree Accuracy	…. (Other methods accuracy)	Naïve Bayes Accuracy
1	80/20	85%		90%
2	80/20	..	..	..
..				

Deliverables for this step:

1. README file indicating which packages you used and any special instructions to compile and run your code.
2. Your code (R is strongly suggested, but you are free to choose any other language)
3. Log file indicating how many experiments you performed before arriving at the best set of parameters.
4. Result file showing comparison of all the classifiers.

V. Analysis:

Write a summary explaining your results. Which method performs best and why do you think it performs the best? Which method is worst and why? Any other analysis that you wish to report. It would be nice to include plots and charts comparing the performance of various classifiers.

Also, compare your performance metrics vs that reported in other relevant papers for this dataset. How do you think your methods fared? What would you change to obtain better accuracy?