## MASTER OF COMPUTER and INFORMATION SCIENCES

COMP809

**Data Mining & Machine Learning**

## ASSIGNMENT ONE

**Semester 1, 2020**

Due**: Friday 3 April at 12 midnight.**

Weighting**: 40%**

**Note: This assignment may be completed individually or in groups of size 2.**

**Submission:** A soft copy needs to be submitted through Turnitin (a link for this purpose will be set up in Blackboard). When submitting the assessment **make the name(s) and student id(s) are indicated on the front page of the report**.

**AIMS**

The Aim of this assignment is two-fold. Firstly, in Part A, you are required to conduct a literature review of data mining applications in Industry, and will thus provide you with a further insight into the ways that data mining is used in practice.

**Part A**

Your survey should cover twodifferent application areas (ensure that these are from different domains – e.g. *banking*, *retail*, *health*, *economics*, etc). The survey is intended to assist you in establishing a suitable framework (application area, tools, algorithms) on which your mining project will be based.

**DELIVERABLES**

* Background information on the organisation that initiated the Data Mining application.
* A brief description of the target application (e.g. detecting credit card fraud, diagnosing heart disease, etc.) and the objectives of the data mining exercise undertaken.
* A description of the data used in the mining exercise (the level of detail published here will differ due to commercial sensitivity, hence flexibility will be used in the marking of this section).
* A description of the mining tools (data mining software) used, together with an identification (no details required) of the mining algorithms *and* how the mining algorithms were applied on the data.
* Discussion of the outcomes and benefits (***be as specific as possible, talk about accuracy of results, potential or actual savings in dollar terms or time savings; do not talk in vague, general terms)*** to the organisation that resulted from the mining exercise. This discussion should contain, in addition to the published material, your own reflection on the level of success achieved by the organisation in meeting their stated aims and objectives.

The total length of your report for Part A is expected to be in the range of 3-4 pages (1.5 to 2 pages for each case study). The criteria that will be used for assessment in Part A is as follows:

|  |  |
| --- | --- |
| **Criterion** | **Mark** |
| **Overall Quality of Presentation** | **4** |
| **Background** | **3\*2=6** |
| **Tools and Mining algorithms** | **4\*2=8** |
| **Outcomes and Benefits** | **6\*2=12** |

**Part B**

Q1)

In this question you will investigate two different methods of combining two different types of classifiers. The dataset that you will be experimenting with is the Forest Type Mapping dataset which can be found from Blackboard under the Assessments 1 folder. The dataset contains information on four different forest types ( 's' ('Sugi' forest), 'h' ('Hinoki' forest), 'd' ('Mixed deciduous' forest), 'o' ('Other' non-forest land) and the objective is to detect (classify) which of the four types of forest that a particular data sample is extracted from. Use 70% of the data for training and the rest for testing.

Use Python for the implementation.

1. The first step is to generate and validate models for each classifier type individually. Use the Decision Tree and the Multi-Layer Perceptron to build individual models.

Display the confusion matrix for each classifier together with the overall classification accuracy generated on the *testing segment*. Submit the Python code for this task. **[3 marks]**

1. Using the models generated in part 1 of this question, generate the probability for *each* class and for *each* classifier and for each sample in the *testing* segment. This should result in 8 probability values for each sample.
2. Submit the Python code for computing the above probabilities. If continuing with the same codebase from part 1 of this question, then highlight the additional code you used. **[5 marks]**
3. Display the probabilities for the *first sample* of the *test segment* of the dataset in a 4 by 2 table where the rows denote class values and the columns denote the type of classifier. **[2 marks]**
4. Using the probabilities generated by the two classifiers for each sample, *formulate a method* of assigning a class for a given sample in the test segment. Your method *must* be based on the *Average* Aggregate function.

Your method must be given *in pseudo code form* (Please ensure that you use *high level English statements* and *not Python-like statements*). [**6 marks]**

1. Implement the method that you used in part 3 of this question in Python code. Submit the Python code that you developed. Run the method and produce the classification accuracy on the *test segment*. [**6 marks]**
2. In this part we will explore a different approach to combining the two classifiers. Instead of performing an aggregation that is based on the *Average* operator we will use conditional probabilities.

With this method we will multiply the maximum probability returned by a classifier by its conditional probability.

Thus for example if the maximum probability across the four different classes returned by the Decision Tree classifier for a given sample was 0.6. Suppose that it was returned for class ‘s’. We then compute P1=0.6\*P(class=s|DT=s).

At the same time if the maximum probability (across the classes) returned by the MLP classifier for that same sample was 0.7 and it was for class ‘h’, then we compute P2=0.7\*P(class=’h’|MLP=h’).

The final class returned by the combined classifier for the sample would be ‘s’ if P1>P2 else it would be ‘h’.

1. Describe an efficient method for evaluating the conditional probabilities required for P1 and P2 above. **[3 marks]**
2. Give one advantage of this method over the method that you proposed in part 3 of this question. **[3 marks]**
3. Implement the method that you used in part 5 of this question in Python code. Submit the Python code that you used. **[7 marks]**

Q2)

In this question you will explore different architectures for building a neural network. The first dataset that you will use here will be the Diabetes dataset which can be found from [here](https://gist.github.com/ktisha/c21e73a1bd1700294ef790c56c8aec1f). Once again use 70% of the data for training and the rest for testing.

1. Use the sklearn.MLPClassifier with default values for parameters and a single hidden layer with k=20 neurons. Use default values for all parameters other than the number of iterations which should be set to 150. Also, as it standard for a MLP classifier, we will assume a fully connected topology, that is, every neuron in a layer is connected to every other neuron in the next layer.

Record the classification accuracy as it will be used as a baseline for comparison in later parts of this question. **[3 marks]**

1. Enable the loss value to be shown on the *test segment* and track the loss as a function of iteration count. You will observe that even when the loss value decreases the error value actually increases between consecutive iterations. Conversely, it is possible that that the error value decreases when the loss increases between consecutive iterations. How do you explain this?

**[4 marks]**

1. We will now experiment with two hidden layers and experimentally determine the split of the number of neurons across each of the two layers that gives the highest classification accuracy. In part 1 of the question we had all k neurons in a single layer. In this part we will transfer neurons from the first hidden layer to the second iteratively in step size of 1. Thus for example in the first iteration, the first hidden layer will have k-1 neurons whilst the second layer will have 1, in the second iteration k-2 neurons will be in the first layer with 2 in the second and so on. Summarise your classification accuracy results in a 20 by 2 table with the first column specifying the combination of neurons used (e.g. 17, 3) and the second column specifying the classification accuracy. **[8 marks]**
2. From the table created in part 3 of this question you will observe a variation in accuracy with the split of neurons across the two layers. Give explanations for some possible reasons for this variation. **[4 marks]**
3. By now you must be curious to see whether the trends that you noted in part 4 above hold true for other datasets as well. Identify *four suitable criteria* for selection of other datasets for further experimentation as described in part 3 of this question. *These criteria must be based on metadata properties*.

**[4 marks]**

1. Source four different datasets that meet your criteria from the UCI Machine Learning repository or Kaggle or any other public source. Briefly describe how these datasets fit the criteria that you identified in part 5. **[4 marks]**
2. Now experiment with each of the 4 datasets that you identified and produce 4 different versions of the 20 by 2 table that you produced in part 3 of this question. Compare the tables that you produced in part 6 with the one in part 3.
3. Discuss whether the trends that you identified in part 4 are also true for the new datasets. [**3 marks]**
4. Give reasons for any difference in trends. **[5 marks]**