F21DL Data Mining and Machine Learning: Coursework 1

Handed Out: Monday 30th September 2019.
Work organisation: group work, in groups of 3 students.
What must be submitted: A report of maximum 4 sides of A4 (five sides of A4 for Level 11), in PDF format, and accompanying software.
Submission deadline: 15:00pm Monday 11th November 2019 -- via Vision
Worth: 25% of the marks for the module**.**

The point:

Data preparation and analysis, confusion matrices, correlation and feature selection are all important in real-world machine learning tasks. Data clustering and probabilistic data analysis are two core sets of methods in data mining and machine learning. So this coursework gives you experience with each of these things.

The data set:

The data set for the coursework is a sample from Stallkamp et al's *German Street Sign Recognition Benchmark*. Originally the data set consisted of 39,209 RGB-coloured train and 12,630 RGB-coloured test images of different sizes displaying 43 different types of German traffic signs. These images are not centred and are taken during different times of the day.

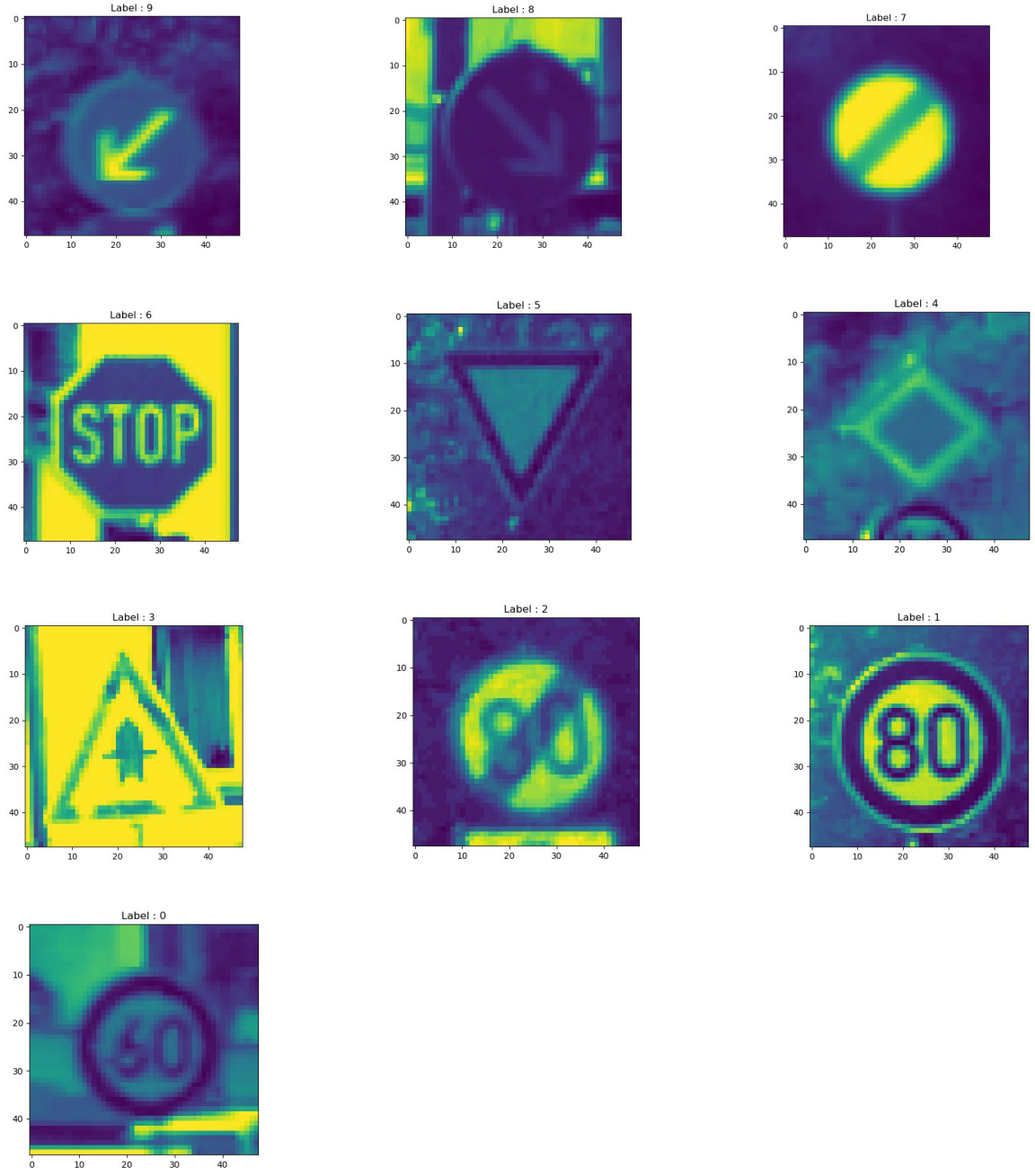
This data set is considered to be an important benchmark for Computer Vision, as has close relation to the street sign recognition tasks that autonomous cars have to perform. And safe deployment of autonomous cars is the next big challenge that researchers and engineers face.

You will be working with a sample of this data set which consists of 10 classes and 12660 images. The images have been converted to grey-scale with pixel values ranging from 0 to 255, and were rescaled to a common size of 48*48 pixels. Hence, each row (= feature vector) in the data set has 2305 features, and represents a single image in row-vector format (2304 features) plus its associated label. Compensating the light conditions and position of the images is not necessary for the coursework and is left for the interested student to do.

Below, the class labels and their meanings are displayed:

0. **speed limit 60** (original label: 3)
1. **speed limit 80** (original label: 5)
2. **speed limit 80 lifted** (original label: 6)
3. **right of way at crossing** (original label: 11)
4. **right of way in general** (original label: 12)
5. **give way** (original label: 13)
6. **stop** (original label: 14)
7. **no speed limit general** (original label: 32)
8. **turn right down** (original label: 38)
9. **turn left down** (original label: 39)

Below are examples of images of the street signs for classes 9 -- 0 in this data set:



For this coursework, we provide 11 training data sets which can be downloaded here: http://www.macs.hw.ac.uk/~ek19/data/. The naming convention is as follows:

1. One entire sample:
   - **[train_gr_smpl.csv]** contains train features and labels from the entire sample represented as row vectors. Class labels range from 0 to 9.
2. Ten one-vs-rest samples:
   - [**train_smpl_<label>**] Train features and labels for one-vs-rest classification. Images with class <label> have a 0 and all other images a 1. For example, if <label> is 6, then all images in the train set displaying a stop sign have a 0 as the label and all other images have a 1.

What to do:

Form or join a group in which you will work; discuss with the group your strategy for completing the courseworks: the workload split, the tools, the methods… Please use Vision **F21DL_2019-2020: Data Mining and Machine Learning** (subpage Assessment) to register your group or join an existing group.

Choose the software in which to conduct the project. We strongly recommend all students use Weka. Weka is a mature, well-developed tool designed to facilitate mastery of machine-learning algorithms. It is supported by a comprehensive textbook: http://www.cs.waikato.ac.nz/ml/weka/book.html . Weka supports embedded Java programming, and you are welcome to use embedded programming in this assignment as it will allow you to automate parts of this assignment. (See the chapter ``Embedded Machine learning in www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf). Alternatively, the Weka command line interface may be embedded inside Bash (shell) scripts, instead of Java.

Students wishing to complete the below tasks in other languages, such as R, Matlab, Python are welcome to do so, assuming they have prior knowledge of these languages. **For students who already know Python, there will be tutorials covering some of the Python libraries for machine learning.**

In the below task specification, the assumption is made that you are using Weka. Please adapt the below instructions accordingly if you use a different programming language.

After collecting the files as above, you will:

1. *[Data Conversion]* Convert all csv files into arff format suitable for Weka. Some suitable data set pre-processing will be needed before you can load the csv file to Weka.
2. *[Data Randomisation]* Produce versions of these files that have the instances in a randomised order.
3. *[Reducing the size, dealing with computational constraints]* The given files may be too big for standard settings of GUI Weka: decide how you are going to deal with this:

- You may reduce the number of attributes, as taught during the course. Record and explain all choices made when you perform the reduction of attributes. A number of algorithms and options are available in Weka. See Sections 2.1 -2.3 in www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf
- Alternatively, you may use the full data set and the Weka command-line interface. See Section 5 of www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf) and manipulate the heap size (see https://weka.wikispaces.com/OutOfMemoryException).
- Either choice is acceptable as long as you can perform the next task.

4. *[Classification: Performance of the Naive Bayes algorithm on the given data set]* Run the Naive Bayes tool in Weka on the resulting version of **train_gr_smpl**. To be able to do this, you may need to apply several Weka "Filters". Explain the reason for choosing and using these filters. Once you can run the algorithm, record, compare and analyse the classifier's accuracy on different classes (as given by the Weka Summary and the confusion matrix).
5. *[Deeper analysis of the data: the data is split into 10 classes, search for important attributes for each class]* For each **train_smpl_<label>** file:

- Using the Weka facility "Select Attributes" for each of these 10 files, record the first 10 fields, in order of the absolute correlation value, for each street sign.

6. *[Try to improve the classification, based on information from item 5]* Using the information about the top correlating features obtained in item (5), transform the full data set **train_smpl** so as to keep the following attributes:
- Using only the top 2 non-class fields from each **train_smpl_<label>**.
- Using only the top 5 non-class fields from each **train_smpl_<label>**.
- Using only the top 10 non-class fields from each **train_smpl_<label>**.
- You will have three data sets, with 14, 35 and 70 non-class attributes respectively. Repeat the experiment described in item (4) on these three data sets.

7. *[Make conclusions:]* What kind of information about this data set did you learn, as a result of the above experiments? You should ask questions such as: Which streets signs are harder to recognise? Which street signs are most easily confused? Which attributes (fields) are more reliable and which are less reliable in classification of street signs? What was the purpose of Tasks 5 and 6? What would happen if the data sets you used in Tasks 4, 5 and 6 were not randomised? What would happen if there is cross-correlation between the non-class attributes? You will get more marks for more interesting and ``out of the box" questions and answers. Explain your conclusions logically and formally, using the material from the lecture notes and from your own reading to interpret the results that Weka produces.
8. [*Beyond Naïve Bayes: complex Bayesian Network Architectures*] Build two or three Bayes networks of more complex architecture for (a smaller version of) this data set, increasing the number of connections among the nodes. Construct one of them semi-manually (e.g use K2 algorithm and vary the maximum number of parents), and two others – using Weka's algorithms for learning Bayes net construction (e.g. use TAN or Hill Climbing algorithms). Run the experiments described in items 4-6 on these new Bayes network architectures. Record, compare and analyse the outputs, in the light of the previous conclusions about the given data.
9. [*Make conclusions*] What kind of new properties and dependencies in the data did you discover by means of using the complex Bayesian Network Architectures? Does it help, and how, to use Bayes nets that are more sophisticated than Naïve Bayes nets? (You may want to read Chapter 6.7, pages 266-270 and pages 451-454 of the Data Mining textbook by Witten et al. before you do these exercises or https://www.cs.waikato.ac.nz/~remco/weka.bn.pdf.)

10. [*Clustering, k-means*] Cluster the data sets **train_smpl**, **train_smpl_<label>** (apply required filters and/or attribute selections if needed), using the k-means algorithm:
- first excluding the class attribute (use *classes to clusters* evaluation to achieve this). This will emulate the situation when the learning is performed in unsupervised manner.
- then including the class attribute. This will emulate the general data analysis scenario.

11. [*Make conclusions*] about the results, compare with classification results obtained in items (1-2).
12. *[Beyond k-means, tools for computation of optimal number of clusters]* Try different clustering algorithms. Try also to vary the number of clusters manually and then use Weka's facilities to compute the optimal number of clusters. Explore various options in Weka that help to improve clustering results. Use the visualisation tool for clustering to analyse the results.
13. [*Make conclusions*] Make conclusions on the obtained improvements to clustering results. Make sure you understand the various details of Weka's output for different (hard and soft) clustering algorithms when clustering is completed. Use Weka's facilities to test the precision of clustering on this data set. Using your work with Weka as a source, explain all pros and

cons of using different clustering algorithms on the given data set. Compare to the results of Bayesian classification on the same data set.

Level 11 only (MSc students and MEng final year students):

14. *[Research Question]* Think about your own research question and/or research problem that may be raised in relation to the given data set, and the topics of Bayesian learning and Clustering. Formulate this question/problem clearly, explain why it is of research value. The problem may be of engineering nature (e.g. how to improve automation or speed of the algorithms), or it may be of exploratory nature (e.g. something about finding interesting properties in data), -- the choice is yours.
15. *[Answer your research question]* Provide a full or preliminary/prototype solution to the problem or question that you have posed. Give logical and technical explanation why your solution is valid and useful.

An Important note:
Before you start completing the above tasks, create folders on your computer to store software you produce, classifiers, Weka settings, screenshots and results of all your experiments. Archive these folders and submit via Vision or via a repository link. As part of your coursework marking, you may be asked to re-run all your experiments in the lab or show the trace of your work (remember, this assignemnt is worth a quarter of your overall module mark!). So please store all of this data safely in a way that will allow you to re-produce your results on request.

What to Submit

You will submit:

(a) All evidence of conducted experiments: data sets, scripts, tables comparing the accuracy, screenshots, etc. Supply a link to your HW web space, github or Google drive.
(b) A report of maximum FOUR sides of A4 (11 pt font, margins 2cm on all sides) for Honours BSc students and FIVE sides of A4 (11 pt font, margins 2cm on all sides) for MSc students.
(c) Your estimate of how much each group member contributed: please be honest and declare if some group members were unable to contribute sufficiently.

Marking: See Rubric on Vision. Maximum points possible: 100.

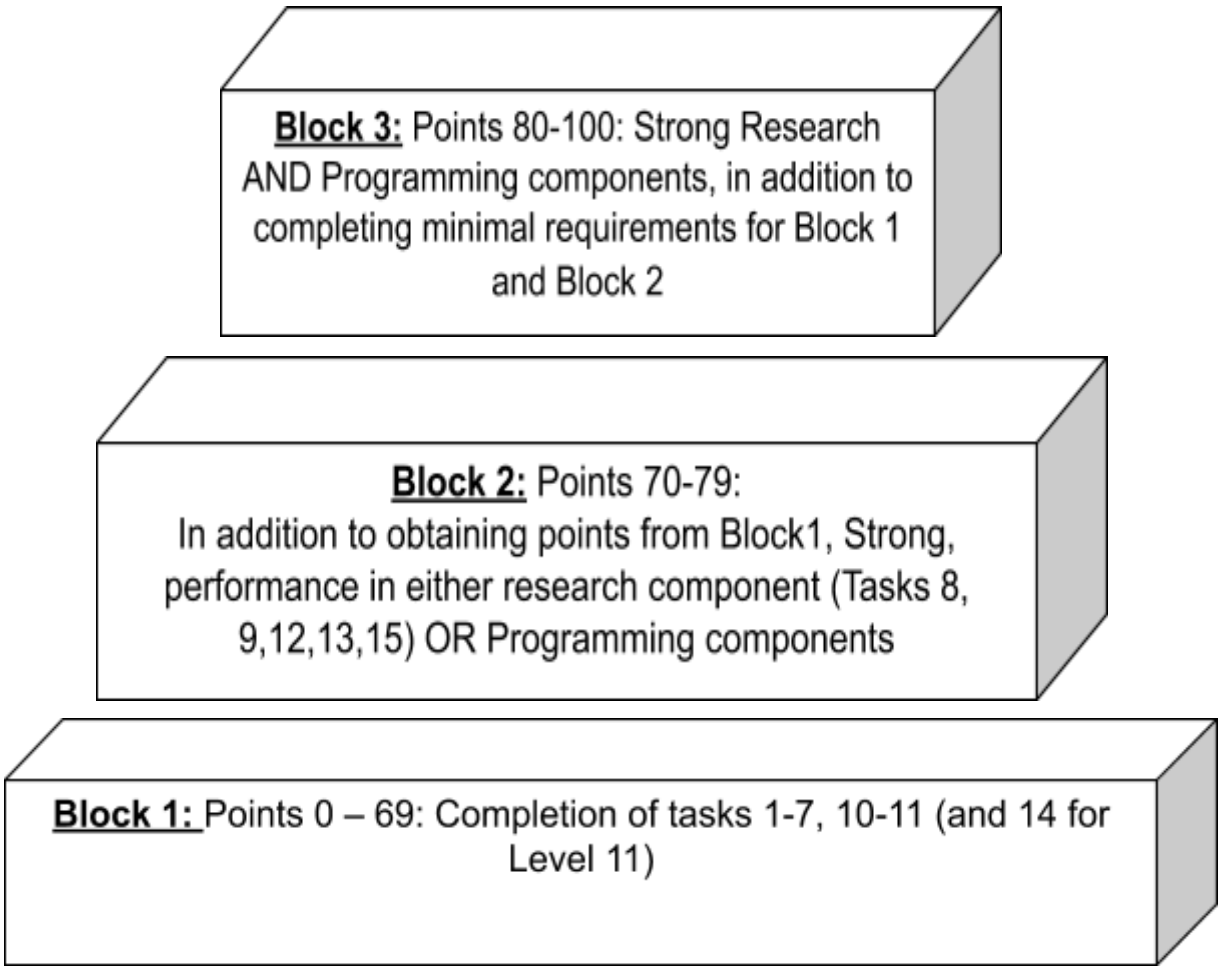
You will get up to 69 points (up to B1 grade) for completing the Tasks 1-7, 10-11 (and Task 14 for **Level 11**) well and thoroughly.

In order to get an A grade (70 points and higher), you will need to first score 69 points as described above, and in addition, you will need to show substantial skill in either research or programming:

- Research skills: Higher marks will be assigned to submissions that show original thinking and give thorough, logical and technical description of the results that shows mastery of the tools and methods, and understanding of the underlying problems. The student should show an

ability to ask his/her own research questions based on the CW material and successfully answer them.

- Programming skills: You will need to produce a sizeable piece of software produced to automate some tasks.
- The mark distribution will thus follow the below scheme:



Plagiarism

This project is assessed as **group work**. You must work within your group and not share work with other groups. Readings, web sources and any other material that you use from sources other than lecture material must be appropriately acknowledged and referenced. Plagiarism in any part of your report will result in referral to the disciplinary committee, which may lead to you losing all marks for this coursework and may have further implications on your degree.
https://www.hw.ac.uk/students/studies/examinations/plagiarism.htm