

Assignment 2

Classification using kNN

For this assessment, you are asked to perform classification using k-nearest neighbour (kNN) on a real-world dataset. The assessment will be marked partly by machine¹. It is therefore very important that you follow the instructions given to you very carefully. Failure to do so will result in a substantial loss of marks, **even possibly zero**.

To help you complete the assessment, you can utilise the program codes that you develop as part of your class exercises.

In the assignment, you will perform the classification of alcoholics vs controls using brain data (EEG) as in the paper² but using the k-Nearest Neighbour (kNN) classifier. You will also further discuss your ideas to improve the classification performance. See below for more details on what is required for submission.

The EEG data was collected from 61 active channels (electrodes, see the reference paper for more information on the dataset but this is not required to complete this submission) where each channel was used to compute a feature. There are 400 patterns from 40 subjects (roughly similar numbers of alcoholics and controls). The 400 pattern is divided into 2 sets:

- Training set: 200 patterns (given in `train_data.txt` file)
- Test set: 200 patterns (given in `test_data.txt` file)

Each row in the file consists of 61 feature values representing either an alcoholic or control subject data. The class labels for the training and test patterns i.e. either alcoholic or control are given as 0 and 1, respectively. *Each of you will have a **different** data set, as such submission outputs will be different (except by chance).*

You will find these datasets (i.e. **four** files):

```
train_data.txt
test_data.txt
train_label.txt
test_label.txt
```

```
in /courses/comp8250/xyz
```

(or equivalent `\\raptor.kent.ac.uk\exports\courses\comp8250\xyz`) on raptor where `xyz` is to be replaced by your own personal login. If you do not find a folder with your login or the folder does not have the required files, then you need to inform me as soon as possible so that the problem can be rectified.

It is reiterated that each student has a different dataset.

¹ Manual inspection will be done to ensure there are no issues such as plagiarism or hard coding and to match the descriptions given in the report.

² R. Palaniappan, P. Raveendran and S. Omatu, "VEP optimal channel selection using genetic algorithm for neural network classification of alcoholics," IEEE Transactions on Neural Networks, pp. 486-491, vol. 13, issue 2, 2002.

Submission

Part A (5 marks)

Your task is to implement kNN algorithm for the classification. You will need to do the following:

- Copy all the data files from the `course` folder to another working folder (or you could also work on the submission `proj` folder).
- Create a java file called, `kNN1.java`
- Load all the data (similar to the class work, after setting the required declarations)
- Implement the necessary codes to compute **Euclidean** distance measures of the test data from the training data, to obtain the predicted labels (using **k=1**) and the classification accuracy.
- Compile and run. As the test data labels are provided, you should be able to obtain the classification accuracy. This will be the benchmark accuracy that you will need to improve for Part B.
- The predicted labels of the test data should also be generated in the `output1.txt` file in the manner prescribed below (write the codes for this). In the first line, there will be a **single line of 200 values** in the file with either **0 or 1** representing the predicted class of the test data with **a single space** between the predicted labels.
- You may wish to compile and run your `kNN1.java` to ensure that it **runs correctly on raptor** as markers will re-compile and run this `kNN1.java` which will generate the predicted labels in `output1.txt`. Using the predicted labels in `output1.txt`, markers will obtain the classification accuracy to be used in the marking scheme.
- You need to place your solution file:
`kNN1.java`

in: `/proj/comp8250/ga/xyz`

(or `\\raptor.kent.ac.uk\exports\proj\comp8250\ga\xyz`) on raptor where `xyz` is to be replaced by your own personal login³. Permissions have been set so that only `xyz` can access files in the directory `xyz`. You will lose write permission at 23:55 on the day of the deadline.

- You should also copy the four data files to the `proj` folder (where you will make the submission) as markers will require these files to assess your submission.
- *It is reiterated that the submission folder is different from the source folder!*

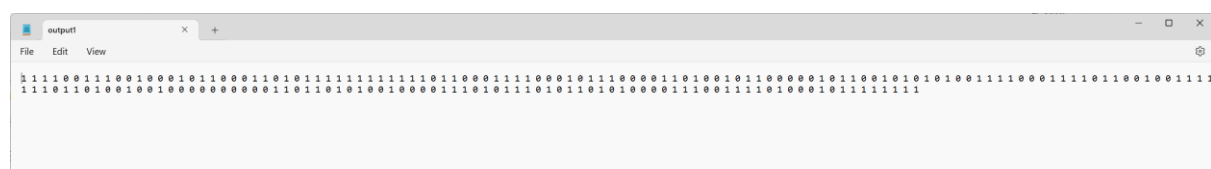


Figure 1: Example of `output1.txt` file contents (there should be a total of 200 binary values with a space between the values). Nothing else should be written to this file.

³ Note that this submission folder is different to the source data folder.

Part B (10 marks)

As a next task, you will implement codes in `kNN2.java` file to improve the classification accuracy (you could reuse any codes in your `kNN1.java` to start with). There are no methods prescribed⁴ but as a starting step, you could utilise the suggestions mentioned in the lectures and additionally by implementing any other appropriate measures.

- As the test data labels are provided, you should be able to explore different approaches (even a combination of approaches) and modify them as necessary using the classification accuracy values as guide.
- There is no need to include codes for failed (poor performing) approaches. You need to include codes only that result in improved classification accuracy as compared to the benchmark accuracy that you have obtained in Part A.
- You will also need to include codes to generate the **final** predicted labels in the `output2.txt` file in the manner prescribed earlier, i.e. in the first line, there will be a single line of 200 values in the file with either 0 or 1 representing the predicted class of the test data with a single space between the predicted labels.

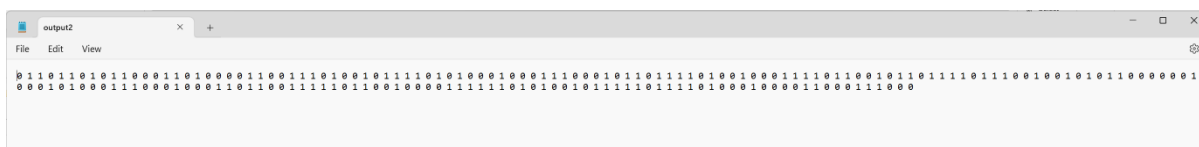


Figure 2: Example `output2.txt` file contents, for Part B (there should be a total of 200 binary values with a space between the values). Nothing else should be written to this file.

- Again, you may wish to compile and run your `kNN2.java` to ensure that it runs correctly on raptor as markers will re-compile and run this `kNN2.java` which will generate the predicted labels in `output2.txt`. Using the predicted labels in `output2.txt`, the markers will obtain the classification accuracy to be used in the marking scheme.
- You need to place your solution file:
`kNN2.java`

```
in:/proj/comp8250/ga/xyz
```

(or `\\raptor.kent.ac.uk\exports\proj\comp8250\ga\xyz`) on raptor where xyz is to be replaced by your own personal login⁵.

⁴ As a hint, there are 10 channels (i.e. 10 columns) that are noisy (i.e. poor features). If these are identified and removed, then it will allow significant increase in accuracy. However, other simpler measures **might** be sufficient to increase the accuracy to obtain the full marks for this task.

⁵ It is reiterated that this submission folder is different from the source data folder.

Part C (10 marks)

You need to describe the methods that you utilised to improve the classification accuracy in Part B. It is not necessary to describe any failed/poor performing approaches, only the approaches that are in your final `kNN2.java` file. You could also include brief details of approaches that you could not implement but have the potential to improve the classification accuracy. The explanation should be in a PDF format up to a maximum of **two A4 pages only** (page limit to include any references), use filename `xyz.pdf`, where `xyz` is your login. You must include sufficient details such as equations, figures, etc to allow the markers to understand the approaches that you took to improve the accuracy as otherwise, you may get fewer marks (or zero) even if your **code runs correctly**.

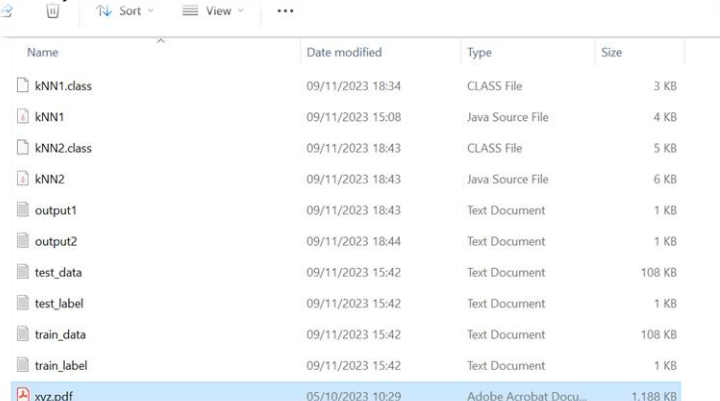
Marking scheme - 25 marks for this assignment are as follows: Your test data output labels will be generated **by markers** and will be matched for accuracy computation.

- Part A: Upto 5 marks will be allocated if your benchmark accuracy matches the expected accuracy⁶ that **should be** obtained with $k=1$ and Euclidean distance.
- Part B (up to a maximum of 10 marks):
 - If no accuracy improvement, mark=0
 - If $0\% < \text{accuracy improvement} < 15\%$, mark=5
 - If $\text{accuracy improvement} \geq 15\%$, mark = $(\text{improved accuracy} - \text{expected benchmark accuracy}) / 3.0$
- From the report, up to 10 marks are allocated for your descriptions of approaches undertaken in Part B.

Total mark is 25 marks (added from parts A, B, and C). Final mark, if not a whole number, will be rounded up (to your benefit).

To check - If you have completed the tasks correctly, you **must** have these files in the root directory of the submission folder (if you do not submit `kNN1.java` and `kNN2.java`, you may get fewer or zero marks):

`kNN1.java`
`kNN2.java`
`test_data.txt`
`train_data.txt`
`test_label.txt`
`train_label.txt`
`xyz.pdf`



Name	Date modified	Type	Size
kNN1.class	09/11/2023 18:34	CLASS File	3 KB
kNN1	09/11/2023 15:08	Java Source File	4 KB
kNN2.class	09/11/2023 18:43	CLASS File	5 KB
kNN2	09/11/2023 18:43	Java Source File	6 KB
output1	09/11/2023 18:43	Text Document	1 KB
output2	09/11/2023 18:44	Text Document	1 KB
test_data	09/11/2023 15:42	Text Document	108 KB
test_label	09/11/2023 15:42	Text Document	1 KB
train_data	09/11/2023 15:42	Text Document	108 KB
train_label	09/11/2023 15:42	Text Document	1 KB
xyz.pdf	05/10/2023 10:29	Adobe Acrobat Docu...	1,188 KB

Figure 3: An example of how your submission folder **could** look like

Do not include additional files other than what is required. If you tested by compiling on raptor, you may have additional class files in the folder generated during Java compilation, which can be left in the folder. You could also leave `output1.txt` and `output2.txt` here. **Do not create any sub-folders.**

⁶ You are NOT told on what is the benchmark expected accuracy for your data with $k=1$ and Euclidean distance.

Penalties

If you do not submit `kNN1.java` and `kNN2.java` files, you will not get any marks (even if you submit the report for Part C).

If `kNN1.java` does not compile and run on raptor for any reason and/or `output1.txt` is not generated correctly, accuracy cannot be obtained, and manual inspection of the code will be performed and only warrants a maximum mark of 3 for Part A.

If `kNN2.java` does not compile and run on raptor for any reason and/or `output2.txt` is not generated correctly, accuracy cannot be obtained, and manual inspection of the code will be performed and only warrants a maximum mark of 5 for Part B. If you submit the report for Part C for a non-compiling/non-executable `kNN2.java` or non-generation of `output2.txt`, only a maximum mark of 5 could be awarded for the described methods.

Note that manual inspection of the code will still be done even for those compiling and executing correctly to identify any plagiarism or **hard coding** of the solution (and to ensure the marks awarded are appropriate). Any detected cases will be reported for disciplinary action.

Raptor

You will need VPN to access university network folders such as raptor when you are off campus. Please see <https://www.kent.ac.uk/guides/work-and-study-from-off-campus/vpn-off-campus-network-access>. For details on accessing raptor, please see <https://www.cs.kent.ac.uk/systems/guides/newuser.html>.

To compile a `java` file on raptor via command line, you will need to use the `javac` command, eg. `javac kNN1.java` and to run, the `java` command, eg. `java kNN1.java`.

Deadline

The deadline for submission is **8 December 2023, 23.55 pm**.

Section 2 of Annex 9 of the Credit Framework does not allow academic staff to accept coursework submitted after the applicable deadline except in concessionary circumstances. For extensions/late coursework submission requests, refer to <https://moodle.kent.ac.uk/2023/course/view.php?id=87>

A Frequently Asked Questions document on Plagiarism and Collaboration is available at: www.cs.kent.ac.uk/teaching/student/assessment/plagiarism.local. The work you submit must be your own, except where its original author is clearly referenced. Checks will be run on submitted work in an effort to identify possible plagiarism, and take disciplinary action against anyone found to have committed plagiarism. When you use other peoples' material, you must clearly indicate the source of the material.