Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CM2203
Module Title: Informatics
Lecturer: Sylwia Polberg-Riener
Assessment Title: Intermediary Informatics Portfolio
Assessment Number: 2 out of 3
Date Set: 19th of February 2024 (Week 4)
Submission Date and Time: by 18th of April 2024 at 9:30am (Week 9)
Feedback return date: 17th of May

If you have been granted an extension for Extenuating Circumstances, then the submission deadline and return date will be later than that stated above. You will be advised of your revised submission deadline when/if your extension is approved.

If you defer an Autumn or Spring semester assessment, you may fail a module and have to resit the failed or deferred components.

If you have been granted a deferral for Extenuating Circumstances, then you will be assessed in the next scheduled assessment period in which assessment for this module is carried out.

If you have deferred an Autumn or Spring assessment and are eligible to undertake summer resits, you will complete the deferred assessment in the summer resit period.

If you are required to repeat the year or have deferred an assessment in the resit period, you will complete the assessment in the next academic year.

As a general rule, students can only resit 60 failed credits in the summer assessment period (see section 3.4 of the academic regulations). Those with more than 60 failed credits (and no more than 100 credits for undergraduate programmes and 105 credits for postgraduate programmes) will be required to repeat the year. There are some exceptions to this rule and they are applied on a case-by-case basis at the exam board.

If you are an MSc student, please note that deferring assessments may impact the start date of your dissertation. This is because you must pass all taught modules before you can begin your dissertation. If you are an overseas student, any delay may have consequences for your visa, especially if it is your intention to apply for a post study work visa after the completion of your programme.

NOTE: The summer resit period is short and support from staff will be minimal. Therefore, if the number of assessments is high, this can be an intense period of work.

This assignment is worth 30% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;
2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Extensions to the coursework submission date can ***only*** be requested using the Extenuating Circumstances procedure. Only students with ***approved*** extenuating circumstances may use the extenuating circumstances submission deadline. Any coursework submitted after the initial submission deadline without *approved* extenuating circumstances will be treated as late.

More information on the extenuating circumstances procedure and academic regulations can be found on the Student Intranet:

https://intranet.cardiff.ac.uk/students/study/exams-and-assessment/extenuating-circumstances
https://intranet.cardiff.ac.uk/students/study/your-rights-and-responsibilities/academic-regulations

By submitting this assignment you are accepting the terms of the following declaration:

I hereby declare that my submission (or my contribution to it in the case of group submissions) is all my own work, that it has not previously been submitted for assessment and that I have not knowingly allowed it to be copied by another student. I declare that I have not made unauthorised use of AI chatbots or tools to complete this work, except where permitted. I understand that deceiving or attempting to deceive examiners by passing off the work of another writer, as one's own is plagiarism. I also understand that plagiarising another's work or knowingly allowing another student to plagiarise from my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings[1].

[1] https://intranet.cardiff.ac.uk/students/study/exams-and-assessment/academic-integrity/cheating-and-academic-misconduct

Assignment

In this assignment, we are going to use excerpts from the following datasets:

- AI Generated Faces from Generated.Photos
https://generated.photos
- Turath-150K Image Database of Arab Heritage
https://danikiyasseh.github.io/Turath/
- ANIMAL-10N Dataset
https://dm.kaist.ac.kr/datasets/animal-10n/
Song, H., Kim, M., and Lee, J., "SELFIE: Refurbishing Unclean Samples for Robust Deep Learning" In Proc. 36th Int'l Conf. on Machine Learning (ICML), Long Beach, California, June 2019

The assignment is worth 30 points in total and is compromised of the following tasks. The classification scheme as well as the data can be found in the .zip file accompanying this portfolio on Learning Central. You **MUST** use the attached template; if the template uses a different programming language than you want to use, please contact the module leader. **Submissions not using the provided template will not be marked and will result in 0 points.**

You are allowed to use libraries to read and write to files, and to perform image transformations if necessary. However, you are not allowed to use libraries to achieve things that you are asked to do on your own. For example, calling a kNN classifier from a scikit-learn package instead of implementing your own from scratch will yield 0 points.

Additional clarifications concerning this portfolio may be posted on the discussion board on Learning Central, so please remember to check it. Please ensure you read the attached template for further instructions concerning technical details of the tasks.

Task 1 [10] My first not-so-pretty image classifier

By using the kNN approach and three distance or similarity measures, build image classifiers.

- You **must** implement the kNN approach yourself
- You **must** invoke the distance or similarity measures from libraries (it is fine to invoke different measures from one library). **Non-trivial adjustments to a library-invoked measure do not meet the requirements!**
- Histogram-based measures are **not allowed**
- Jaccard distances/similarities are **not allowed**
- You can use between 0 and 3 distance measures and between 0 and 3 similarity measures (there is no requirement that at least one of each kind should be present)

The classifier is expected to use only one measure at a time and take information as to which one to invoke at a given time as input. The template contains a range of functions you must implement and use appropriately for this task.

You can start working on this task immediately. Please consult at the very least Week 2 materials.

Task 2 [4] Basic evaluation

Evaluate your classifiers. On your own, implement a method that will create a confusion matrix based on the provided classified data. Then implement methods that will output precision, recall, F-measure, and accuracy of your classifier based on your confusion matrix. Use macro-averaging approach and be mindful of edge cases. The template contains a range of functions you need to implement for this task.

You can start working on this task immediately. Please consult at the very least Week 3 materials.

Task 3 [6] Cross validation

Evaluate your classifiers using the k-fold cross-validation technique covered in the lectures (use the training data only). Output their average precisions, recalls, F-measures and accuracies. You need to implement the validation yourself. Remember that folds need to be of roughly equal size. The template contains a range of functions you need to implement for this task.

You can start working on this task immediately. Please consult at the very least Week 3 materials.

Task 4 [3] The curse of k

Independent inquiry time! Picking the right number of neighbours k in the kNN approach is tricky. Find a way you could approach this more rigorously. In comments:

- state the name of the approach you could use,
- give a one-sentence explanation of the approach, and
- provide a reference to it (use Cardiff University Harvard style, **DOI MUST BE PRESENT**).

The reference **must be** a handbook or peer-reviewed publication; a link to an online tutorial will not be accepted. Ensure that your resources are respectable and are not e.g., predatory journals.

You can start working on this task immediately. Please consult at the very least Week 2 materials.

Task 5 [4] Similarities

Independent inquiry time! In Task 1, you were instructed to use libraries for image similarity measures. Pick two of the three measures you have used and implement them yourself. You are allowed to use libraries to e.g., calculate the root, power, average or standard deviation of some set (but, for example, numpy.linalg.norm is not permitted). The template contains a range of functions you need to implement for this task.

Disclaimer: if you decide to implement MSE, do not implement RMSE (and vice versa)

You can start working on this task immediately. Please consult at the very least Week 1 materials.

Task 6 [3] I can do better!

Independent inquiry time! There are much better approaches out there for image classification. Your task is to find one, and using the comment section of your project, do the following:

- State the name of the approach
- Provide a permalink to a resource in the Cardiff University library that describes the approach
- Briefly explain how the approach you found is better than kNN in image classification (2-3 sentences is enough). Focus on synthesis, not recall!

You can start working on this task immediately. Please consult at the very least Week 2 materials.

Learning Outcomes Assessed

1. Execute and evaluate various techniques in knowledge discovery and data mining.

Criteria for assessment

Points required for every level are as follows:

- First: >= 70% ; >= 21 points
- Upper second: >= 60%; >= 18 points
- Lower second: >= 50%, >= 15 points
- Pass: >= 40%, >= 12 points
- Fail: < 40%, < 12 points

Coded Tasks

Credit will be awarded against the following criteria. The tasks are connected and increase in difficulty. Parts of this portfolio will be marked automatically, hence it is crucial that you properly prepare your code to work on a different machine. **Code that does not run/compile or where function does not respect input/output types or patterns stated in the function documentation may receive 0 points or a suitable loss of points.**

Task 1	Points possible
Is first measure correctly implemented and handled?	1
Is second measure correctly implemented and handled?	1
Is third measure correctly implemented and handled?	1
Is the read and resize function correct and properly used?	1
Is the validation function correct and properly used?	1
Is the most common class function correct and properly used?	0.5
Is the nearest neighbour function correct and properly used?	1
Is the core kNN function present and correct?	3

Does the code produce appropriate output?	0.5

Task 2	Points possible
Is the core evaluatekNN function present and correct?	0.5
Is the classified data validated?	0.25
Is the confusion matrix function correct and properly used?	0.75
Is the TP function prepared well?	0.3
Is the FP function prepared well?	0.3
Is the FN function prepared well?	0.3
Is macro precision complete and correct and properly used?	0.4
Is macro recall complete and correct and properly used?	0.4
Is macro f-measure complete and correct and properly used?	0.4
Is accuracy complete and correct and properly used?	0.4

Task 3	Points possible
Is the core of the crossEvaluatekNN function present and correct?	1.5
Is the splitting function complete and correct and properly used?	2
Is the evaluation function complete and correct and properly used?	0.5
Is the validation function correct and properly used?	1.25
Does the code produce appropriate output?	0.75

Task 5	Points possible
Is first measure correctly implemented and used?	2
Is second measure correctly implemented and used?	2

Submissions not using the provided template will NOT BE MARKED and will result in 0 points.

All code is expected to be accompanied by appropriate comments that are supposed to help the marker to understand your code. Lack of such documentation can incur penalty points, and staff reserve the right not to mark code that is not understandable due to lack of proper documentation.

All code is expected to be reasonably clean and succinct. Please note that there is no need to aim for the most minimal code possible. However, if you decide to use 100 lines of code to write something that can be done in 10, penalty may be applied.

Written Tasks

Credit will be awarded against the following criteria.

Task 4	Points possible
Is the proposed approach good?	1
Is the reference material good & reputable?	1

Is the citation/referencing style good?	1

Task 6	Points possible
Is the proposed approach good?	1
Is the permalink present and correct?	0.4
How are the explanations? Are they clear, correct, succinct?	1.6

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on the 17th of May via Learning Central and/or email.

Feedback from this assignment will be useful for any other modules requiring programming or machine learning.

Submission Instructions

Description		Type	Name
Source code	**Compulsory**	One .zip folder containing: 1. coded tasks using the template 2. requirements.txt/lib folder as appropriate for the language 3. version.txt file warning of Python/Java version used if it is different from the requested one	[student_number].zip

DO NOT include the image folders or csvs supplied with the template in your submission. Submissions not using the provided template will NOT BE MARKED and will result in 0 points.

The files need to be submitted to appropriate parts in the Assessment→ Portfolio 2 area on Learning Central. Any code submitted will be run on a system equivalent to those available in the Windows laboratory and must be submitted as stipulated in the instructions above.

In all coding tasks, please use one of the following: Java (JDK 1.14 or JRE 1.8) or Python (version 3.8). **If you require a different version, you need to warn the module leader** using the version.txt file as stated above.

Use of Maven, Google Colab and Jupyter Notebook is NOT PERMITTED.

The code needs to be prepared in a way that it can run on a different machine without marker's intervention (Java submission using special libraries should include these libraries, Python submissions should include a requirements.txt file). No path modifications that require manual intervention from the marker can be present (the image paths should be taken from files as-is, and file paths need to be as taken at input).

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part OR a reduction in marks for that assessment or question part.

Staff reserve the right to invite students to a meeting to discuss coursework submissions.

Automatic anti-plagiarism and similarity checking tools can be used to process the submissions.

You can submit multiple times on Learning Central. ONLY files contained in the last attempt will be marked, so make sure that you upload all files in the last attempt.

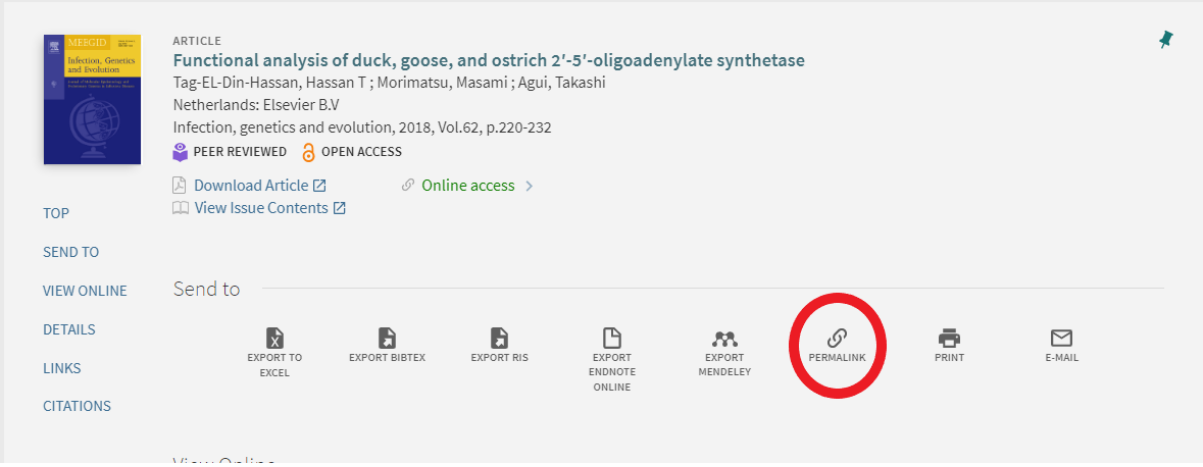
Support for assessment

Please feel free to:

- approach the module leader during face-to-face or optional practicals,
- request office hours,
- send an email to polbergs@cardiff.ac.uk using the previously provided template,
- write on COMSC Discord Server (sylwiathepanda)
- write on the CM2203 Discussion Board on Learning Central

FAQ (Please also consult discussion board)

1. **What do you mean by "implement yourself"?**
   Personally write the code; do not invoke the ready functions to do what I ask for from library or copy paste from online tutorials. Think about it this way; I am asking you to personally bake a cake. It's fine to shop for ingredients at the store, but if you bring me a ready store-bought cake or one baked by your grandma, you will not get any credit.
2. **Where should I put the image folders and csv files?**
   The paths to csv files are provided at input, put them wherever you like. The paths to images in the csv files are relative paths; either put the image folders in appropriate spot in your active project directory, or change the paths in csvs to absolute paths and put the image folder wherever that points to.
3. **Which citing and referencing format should we use?**
   As long as the style is consistent and references include all required elements, I don't care which one you use. For examples and tutorials on some possible styles, visit the following page: https://intranet.cardiff.ac.uk/students/study/study-skills/academic-writing-communication-and-referencing/citing-and-referencing-support
4. **What do you mean by Cardiff University Library permalink?**
   I mean the link you get after pressing the permalink button on the CU library page of a given book/article:



5. **The algorithm is so slow! Will this affect my mark?**
   Nope, I don't care about efficiency, I care about correctness.
6. **There are so many images that testing is really troublesome!**
   Then test on a subset, simply create a smaller csv file and that's that. Welcome to basics of testing and debugging 😊
7. **How do I know if I implemented things right?**
   Testing, testing, testing. Different libraries have different assumptions and handling of variables. Just because your answer is the same as from a given library, it does not mean it's right, since you could have used the library wrong. Might be best to have some simple examples where you are completely confident of the end result (might be best to calculate by hand) and then see if your algorithm returns what you need.
8. **What happens if my code does not have comments explaining what is happening?**
   You may lose marks. I expect all projects to be appropriately documented.
9. **I don't know any similarity measure libraries!**
   Google is your friend. Also, check padlet from Week 1.
10. **Why are you passing functions as input parameters?! That looks complicated!**
    There is one of me, and triple digits of you. I will be supporting myself with automated marking, which means I need to be able to test if the logic of a given function is ok even if the other functions on which it might depend are broken.