Credits: 50%

Deadline: 20/01/2025 4PM (Monday Week 12)

In this Final Assignment, you will practice your skills in programming using *Python*, and in analysing data using *R*.

======== Python ========

Your task will be to implement and evaluate a Decision Tree classifier in Python, and compare it against a library implementation (e.g., sklearn). When evaluating the classifiers, please use datasets from the *UCI Machine Learning repository:* *http://archive.ics.uci.edu/ml/index.php*.

You need to evaluate both computational aspects (e.g., training time, memory, etc.) and machine learning aspects (e.g., accuracy, precision, recall, etc.). As part of your analysis, you should vary the problem size (e.g., size of the training set). You should also change hyper-parameters, and then evaluate how it affects your implementation and the library implementation. For example, changing the maximum depth of the decision tree, changing the criteria for creating leaf nodes, etc.

You will collect this data about the classifiers in Python, save them in intermediate csv files, and submit them as a zip file.

In addition to your Python code, you also need to submit a statement about your work and test cases, as described next.

Statement

In the same folder as your Python files, please include a file "**statement.pdf**" where you answer the following questions:

1.) Is the code your own implementation, an adaptation of existing code, or a direct copy of existing code? Note that, only changing the name of variables, functions and methods is still considered a direct copy.

a) Own implementation
b) Adaptation of existing code
c) Direct copy

If you adapted existing code, how exactly did you modify the code? Did you implement any new features? Which part of the code is your own work?

2.) Please list all the external resources used, and how they were used. An external resource includes existing code, libraries, web pages, books, web forums, friends, etc. Note that this question is required even if you worked on your own implementation.

Note that wrong or missing statements will be handled as described in the plagiarism framework (*see section below*). Mark deduction or zero marks for the project are likely consequences.

Test Cases

You must also include several test cases using the the *pytest* framework. You can read about *pytest* at https://docs.pytest.org/en/7.2.x/getting-started.html. Hence, your folder should include one or more Python files that are prefixed by *"test_"*, which include test functions prefixed by "*test_*". Your Python grade will take into consideration the extension and thoroughness of your tests.

Please include the output of *pytest* in a file called "*test-output.txt*". However, you should also make sure that your tests would run during marking. That is, if you use additional files in your tests, then they must be included in your submission folder.

======== Statistical analysis in R ========

After having created the csv files in Python, we will load them in R and use them for a statistical analysis. This part must be done in R using Rmarkdown, see also the report section below. As said above, you will compare your implementation with the existing library implementation. You will analyse computational and machine learning aspects. You will consider how they depend on the problem size and parameters of your method. We expect an in-depth analysis, using different statistical tools in R. You can use for instance linear regression, t-tests, graphs using ggplot2, etc. Your analysis is not restricted to functions that you have learned in Week 1-5. You are expected to use other methods and functions as well. It is required that you inform yourself about them.

The R code must follow good coding standards and should be modern and easily readable. R code written using "old" (more traditional) R statements/functions that drift away from the tidyverse philosophy will be penalised heavily. This includes reproducibility. Your R Markdown file (.Rmd) must work on our computers and any other. All what we will do for that is to unzip the zip file with the csv files in the same folder as the Rmarkdown file. Make sure that the Python script producing the .csv is **also reproducible and easy to source**. I will test reproducibility of this via sourcing the Python script through a fresh virtual environment under 3.11.X. Make sure all packages install under that version via pip, or are easily accessible.

Additionally, the R code should be annotated, indented, variables and functions should be well named, lines should not be too long, etc. Differently to previous courseworks, in generating the .html, **please do not 'echo' your R code**. You must set echo = FALSE in the chunks. Also, you are more than welcome to use inline R code evaluations (read how to do so in the markdown guide). Also, do not rely on print statements. Either create plots, perhaps small tables, but figures are often better, or use inline statement for results `r variableName`. **Assume your .html report should be readable by a manager that does not know how to code**, however has statistical knowledge.

Similar to the Python part, you are allowed and expected to use external resources, but you have to give references. For statistical methods, please use the reference system. Your final report should contain them. For R code, use comments in your files to clarify any external resource used. Also here, see the plagiarism section below for more details.

======== Report ========

The report must be written using Rmarkdown. It should make clear what you have done in Python and what statistical analysis you have performed in R. Do not use Python or R code to do so. You are expected to describe, explain, interpret, contextualise and summarise your experimental results. It is a good idea to see this project as mini version of your later Master thesis.

Please write your report using the following sections: *Abstract*, *Introduction*, *Methodology*, *Results*, *Discussion*, *Conclusion*, *Acknowledgements (if required)*, *Bibliography*. Some notes:

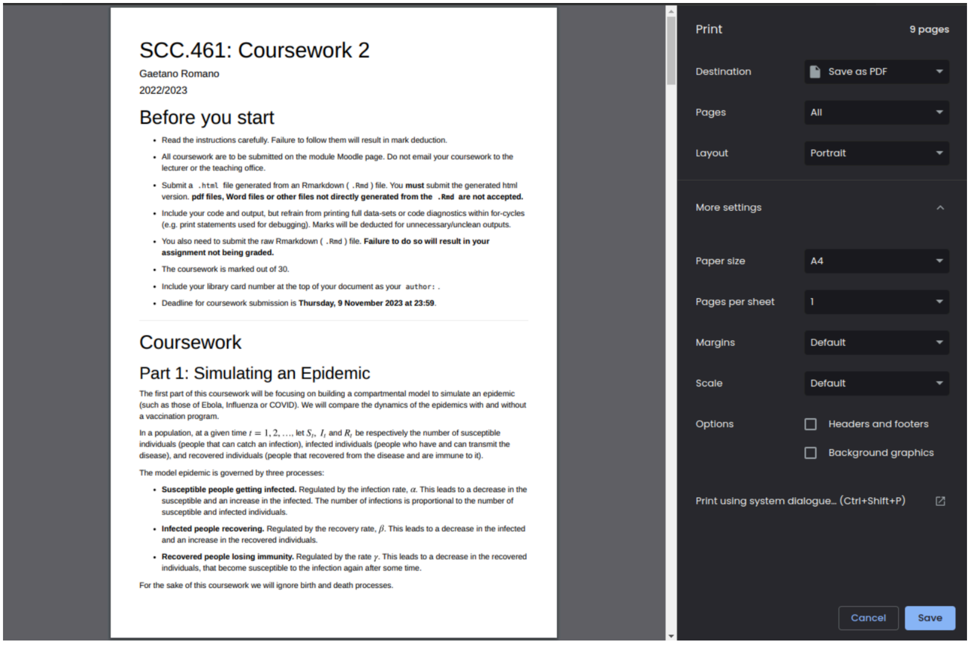
- The Abstract should be no more than 250 words.
- Presentation is important. This includes your writing, spelling, correct use of section headings and so on.
- The report of the analysis in R should be reproducible, that is, the pdf file of your report can be generated from your Rmarkdown source code.
- The Methodology section should explain the Decision Tree algorithm that you have implemented. Please describe all the functionalities that you implement (e.g., ability to handle symbolic features, ability to handle multiple datasets, etc).
- The Discussion section should discuss the implications of your experimental results, and mention potential explanations.

Report Submission.

The report must be submitted in a .pdf.

There are two ways of doing so:

- By directly knitting the Rmarkdown by pdf. This is done by setting your notebook output to .pdf rather then .html. It should work out of the box on the university machines. On personal machines requires a LaTeX installation. If you are not familiar with how to install LaTeX, I recommend either using the university machines, or the following option.
- You can knit your Rmarkdown in html. Then, you open it up in a browser and print the webpage (ctrl+p or cmd+p on mac). Select the A4 format and print to .pdf (*save as PDF in Chrome*). Keep 1 page per sheet, default margins, default scale. Submit the resulting .pdf. See below for an example:



SCC.461: Coursework 2

Gaetano Romano

2022/2023

Before you start

- Read the instructions carefully. Failure to follow them will result in mark deduction.
- All coursework are to be submitted on the module Moodle page. Do not email your coursework to the lecturer or the teaching office.
- Submit a `.html` file generated from an Rmarkdown (`.Rmd`) file. You **must** submit the generated html version. **pdf files, Word files or other files not directly generated from the `.Rmd` are not accepted.**
- Include your code and output, but refrain from printing full data-sets or code diagnostics within for-cycles (e.g. print statements used for debugging). Marks will be deducted for unnecessary/unclean outputs.
- You also need to submit the raw Rmarkdown (`.Rmd`) file. **Failure to do so will result in your assignment not being graded.**
- The coursework is marked out of 30.
- Include your library card number at the top of your document as your `author:` .
- Deadline for coursework submission is **Thursday, 9 November 2023 at 23:59**.

Coursework

Part 1: Simulating an Epidemic

The first part of this coursework will be focusing on building a compartmental model to simulate an epidemic (such as those of Ebola, Influenza or COVID). We will compare the dynamics of the epidemics with and without a vaccination program.

In a population, at a given time $t = 1, 2, \ldots$, let S_t, I_t and R_t be respectively the number of susceptible individuals (people that can catch an infection), infected individuals (people who have and can transmit the disease), and recovered individuals (people that recovered from the disease and are immune to it).

The model epidemic is governed by three processes:

- **Susceptible people getting infected.** Regulated by the infection rate, α. This leads to a decrease in the susceptible and an increase in the infected. The number of infections is proportional to the number of susceptible and infected individuals.
- **Infected people recovering.** Regulated by the recovery rate, β. This leads to a decrease in the infected and an increase in the recovered individuals.
- **Recovered people losing immunity.** Regulated by the rate γ. This leads to a decrease in the recovered individuals, that become susceptible to the infection again after some time.

For the sake of this coursework we will ignore birth and death processes.

Print 9 pages

Destination: Save as PDF

Pages: All

Layout: Portrait

More settings

Paper size: A4

Pages per sheet: 1

Margins: Default

Scale: Default

Options: □ Headers and footers □ Background graphics

Print using system dialogue... (Ctrl+Shift+P)

Maximum Length The output .pdf should be fitting on a maximum of 13 sides of A4 (*excluding Acknowledgements and Bibliography*). Any content outside the 13 pages will not be graded.

Submission

In the submission system, please upload the following files:

- The .pdf report
- The Rmarkdown code that regenerates the report, but renamed to a txt file

- A zip file with the python code, including the "statement.pdf" file, all your test cases, and all csv files (as described above in the Python section)
- A zip file with additional materials that may be needed to re-generate your report.

Template & zip archive

A zip file of templates is provided here . Knitting template_rmarkdown.Rmd will generate template_rmarkdown.pdf. You must general a pdf file (no html or docx). In your submission, you can include accompanying files similar to references.bib and flow_charts.png to ensure reproducibility. Just add a zip file with the additional materials, as described above.

Plagiarism

Plagiarism is presenting someone else's work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgement. All published and unpublished materials, whether in manuscript, printed or electronic form, is covered under this definition. Plagiarism may be intentional or reckless, or unintentional. As usually, it is strictly forbidden and will be penalised. University procedures will be followed. Note that the university runs automatic tools to help us detecting plagiarism. If you look at it, you can see that potential consequences are mark deduction, zero marks, and up to leaving the university without a degree.

You can use any references that you want and re-use existing code. However, you must cite clearly all the references that you use, and all the code that you re-used. You must do "significant work" beyond the code that you re-used. Your grade will reflect the amount of original work that you produce in this project provided that you give the proper references when needed.

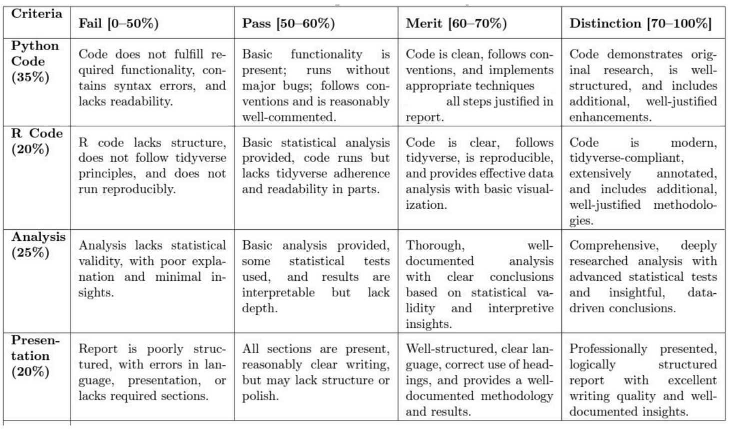
For example, if you fully re-use an existing Python code for the Decision Tree, instead of producing your own Decision Tree, you can expect a low "Python" grade, but you will still be marked concerning the other aspects of this coursework, provided that you clearly gave the proper references. However, if you fully re-use an existing Python code for the Decision Tree, but falsely claim that it is your original work, and/or do not write any references to the original work, then you may get zero marks for the whole project, and could be referred to a plagiarism hearing.

Similarly, you are free to discuss your work with your friends and colleagues, but each student must submit their own work. That is, if you are stuck with a bug, you can ask a friend for help, but you cannot just copy and paste someone else's assignment. You must also report if someone helped you (in the "statements.pdf" file, and in an "Acknowledgements" section in your report). In general, it is a good academic practice and a good work practice to show your gratitude in the Acknowledgements when someone helps you in your project.

Marking Scheme

The marks for this project are awarded as follows:

- Python code - 35%
- R code - 20%
- Analysis (correctness and depth) - 25%
- Presentation (e.g., text quality, usage of plots etc.) - 20%



Criteria	Fail [0–50%)	Pass [50–60%)	Merit [60–70%)	Distinction [70–100%]
Python Code (35%)	Code does not fulfill required functionality, contains syntax errors, and lacks readability.	Basic functionality is present; runs without major bugs; follows conventions and is reasonably well-commented.	Code is clean, follows conventions, and implements appropriate techniques all steps justified in report.	Code demonstrates original research, is well-structured, and includes additional, well-justified enhancements.
R Code (20%)	R code lacks structure, does not follow tidyverse principles, and does not run reproducibly.	Basic statistical analysis provided, code runs but lacks tidyverse adherence and readability in parts.	Code is clear, follows tidyverse, is reproducible, and provides effective data analysis with basic visualization.	Code is modern, tidyverse-compliant, extensively annotated, and includes additional, well-justified methodologies.
Analysis (25%)	Analysis lacks statistical validity, with poor explanation and minimal insights.	Basic analysis provided, some statistical tests used, and results are interpretable but lack depth.	Thorough, well-documented analysis with clear conclusions based on statistical validity and interpretive insights.	Comprehensive, deeply researched analysis with advanced statistical tests and insightful, data-driven conclusions.
Presentation (20%)	Report is poorly structured, with errors in language, presentation, or lacks required sections.	All sections are present, reasonably clear writing, but may lack structure or polish.	Well-structured, clear language, correct use of headings, and provides a well-documented methodology and results.	Professionally presented, logically structured report with excellent writing quality and well-documented insights.

Last modified: Tuesday, 3 December 2024, 16:13