University of Aberdeen

School of Natural and Computing Sciences

Department of Computing Science

MSc in Artificial Intelligence

2024 – 2025

CA1, Briefing Document – Group Work

Title: CS5079 – Applied Artificial Intelligence

Information for Plagiarism

The instructor will submit the source code and the report for plagiarism check. Please refer to the slides available on MyAberdeen for more information about avoiding plagiarism before starting the assessment. Please also read the following information provided by the university:

https://www.abdn.ac.uk/medical/electives/elective_information/page/64 .

https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/.

By submitting your work, every member of the group agrees that they have taken all measures to avoid all kinds of misconduct (plagiarism, collusion, or contract cheating). If any kind of misconduct is detected, the WHOLE group will take responsibility for it.

Learning Objectives

- **Explore a challenging problem, understand its nature, and how it impacts AI model selection and evaluation.**
- **Implement neural networks for reinforcement learning.**
- **Gain the ability to read, understand, and critique state-of-the-art research papers.**
- **Understand the challenges in developing and deploying an online system.**

Guidance

The group will have to upload two distinct elements:

- A PDF document containing their answers to the questions below. Make sure that each answer can be easily matched to the corresponding question.
- The necessary code in one Python notebook (as a ZIP file). ***Do not clear the output of the cells to allow the marker to easily check the correct computation of the code.***

The report should describe and justify each step that is needed to reproduce the results by using code-snippets, screenshots, and plots. When using screenshots or plots generated in Python, make

sure they are readable. If the students used any open-source code, they must point out where it was obtained from (even if the sources are online tutorials or blogs) and detail any modifications they have made to it. The students should mention this both in the code and the report. *Failure to do so will result in academic misconduct sanctions.*

Questions about any aspects of this assessment should be addressed to the course coordinator.

Marking Scheme

The instructor will take the following marking criteria into account:

- **Quality of the report**, including structure, clarity, good English, and brevity.
- **Reproducibility**. How easy is it for another student (of their cohort) to repeat the experiments based on the report and code provided?
- **Understanding**. Do the students show a deep understanding of the approaches used?
- **Quality of the experiments**, including design and the presentation of the results.
- **In-depth analysis of the results**, including critical evaluation and conclusions.
- **Quality of the source code**, including the documentation of the code and comments.

This assignment will be marked out of 100 marks by the instructor. All group members will receive the project mark (exceptions apply, discuss this with the course coordinator). This coursework will be worth 50% of the overall course mark.

Task 1: Reinforcement learning on the FrozenLake-v1 environment (50 marks)

This assessment will focus on the **FrozenLake-v1** environment. In this environment, the player guides an agent in a grid with frozen surfaces and holes to retrieve a chest. Some tiles are slippery which can cause the player to move in an undesirable direction. For more details, see the GitHub page: https://github.com/openai/gym/blob/master/gym/envs/toy_text/frozen_lake.py

Figure 1: Screen of the FrozenLake-v1 game.

For this task, the students are required to generate a random 10x10 valid grid with **a 30% probability that a tile is frozen.** Use the provided notebook ("submission.ipynb") to initialize the environment.

DO NOT PUT THE "is_slippery" PARAMETER TO FALSE OR THE TOTAL MARK WILL BE REDUCED.

The group can make use of the provided YML file ("environment.yml") to initialise their Anaconda environment. Please use Python for all programming tasks. The students are encouraged to use python-based frameworks, such as **Tensorflow and Keras**.

1.1) Describe, in detail, the following game elements using your own words. (**2.5 marks**):

- Observations
- Action space
- Reward
- The environment's info dictionary
- Episode

1.2) Implement an agent based on a neural network, using a parameter ε, for making random moves, that decreases from 1 to 0.1. Describe how you deployed your agent, the motivation behind the design choices, and how you adjusted its parameters, going into detail on what each parameter does as well. You may use open-source code and libraries if you acknowledge them (**15 marks**).

1.3) Train the agent created in 1.2 on the game. *Please note that a high number of episodes may be required for the agent to reach the goal depending on your implementation.* Present the training process, the experiments (including the experimental setting), and discuss your results. You should make use of figures, including a line plot that shows how the average amount of rewards over episodes evolves over time (**5 marks**).

1.4) Randomly relying on the exploration of the grid can be a time-consuming process. Implement another agent, using **ANY** technique of your choice, which is aware of another additional information: *The position of the chest at the bottom right corner.* The agent should not cheat, i.e., they should not (at least initially) be aware of the layout of the map, i.e., the positions of the holes. Explain all design choices that were made to create this agent (**20 marks**).

1.5) Evaluate the agent created in 1.4 and discuss your results with respect to the previous questions (**5 marks**).

1.6) Upload a video (maximum 2 minutes) of your best agent starting from the start position (at the top left) and reaching the goal (at the bottom right). If a link to the video is not provided, you will receive 0 marks for this sub-question (**2.5 marks**).

Task 2: Research and Advances in Bias in Recommendations (20 Marks)

Review papers from the International Workshop on Algorithmic Bias in Search and Recommendation (BIAS) 2023.

2.1) Carefully select and choose 2 papers from the workshop proceedings (https://dblp.org/db/conf/bias/bias2023.html) and, for each paper, use your own words to describe its context, problems, and contributions (400 words maximum per paper) **(10 marks).**

2.2) Discuss and critically evaluate the papers described in 2.1 (500 words maximum per paper). You should answer important questions, including: What is the type of paper? How well are the research questions addressed? Is the approach realistic? **(10 marks).**

Task 3: Creating a Recommendation Engine for Songs (30 Marks)

3.1) Download the "song_dataset.csv" dataset from Blackboard and import it. The column descriptions are as follows:

- "user" is the ID of the listener
- "song" is the ID of the song listened
- "listen_count" is the number of times the song was listened by the user.
- "title" is the name of the song
- "release" is the name of the album
- "artist_name" is the name of the artist
- "year" is the release year.

Perform the necessary exploratory data analysis and discuss your findings. You should make use of graphs and tables, as necessary. The submission should include answers to the following questions:

- What are the most listened songs?
- Who are the most popular artists?
- How is the distribution of song count for users?

(7.5 marks)

3.2) Implement a recommendation engine which takes as input a user in the database and recommends a song he has not listened to yet. Describe your approach and briefly discuss your design choices and what the other possible alternatives are. **(15 marks)**

3.3) Deploy your recommendation engine on a platform of your choice. The user should be able to input the songs they have already listened to (using drop-down menus) and the platform should provide recommendations. The answer to this sub-question should include a working URL to the platform. **(7.5 marks)**