1. The Language Classification Problem:

This outlines a machine learning problem in which you are required to create a model that can discriminate between English, Afrikaans, and Dutch phrases. A labelled dataset of phrases is provided/attached (lang_data.csv).

Requirements:

Implementation

- You must implement your machine learning model in Python.
- You are free to use external Python libraries, but pre-trained models may not be used.
- You will have to submit your code, as well as a trained model.
- Include instructions for executing your code with the provided (trained) model.

Report

- In addition to the code and model, you must also submit a report. The report should contain at least the following:
    - a brief overview of the libraries used in your implementation,
    - a complete overview of any data analyses, preparation, and/or feature extraction that you performed,
    - a complete overview of your model's architecture,
    - a complete overview of the training process, including detailed discussions of any specific techniques and/or algorithms used in your implementation,
    - an overview of the testing process,
    - an overview and discussion of the results, including your model's performance on each language, and how it can potentially be improved..
- Where applicable, justify any choices you make in your approach.
- Also answer the bonus questions (see below) in your report.

Bonus Questions:

To improve your chances, answer the following questions in your report. You may include these answers as part of discussions throughout your report, or answer them directly in a separate section.

1. Discuss two machine learning approaches (other than the one you used in your language classification implementation) that would also be able to perform the task. Explain how these methods could be applied instead of your chosen method.
2. Explain the difference between supervised and unsupervised learning.
3. Explain the difference between classification and regression.

4. In supervised classification tasks, we are often faced with datasets in which one of the classes is much more prevalent than any of the other classes. This condition is called *class imbalance.* Explain the consequences of class imbalance in the context of machine learning.
5. Explain how any negative consequences of the class imbalance problem (explained in question 4) can be mitigated.
6. Provide a short overview of the key differences between deep learning and any non-deep learning method.

2. Pong

This outlines two machine learning problems which will require you to train game-playing agents from zero-knowledge.

Please choose and complete one of the two options outlined in this report. Regardless of which option you choose, the requirements below must be met in your submission.

Requirements

Implementation

- Choose either option A or B below.
- You must implement your game and machine learning model in Python.
- You are free to use external Python libraries.
- There is no graphics requirement. If you choose to implement an ascii game, you will not be penalised.
- You will have to submit your code, as well as a trained game-playing agent.
- Include instructions for executing your code with the provided (trained) game-playing agent.
- The game overviews in options A and B are not set in stone. You may modify them if you wish to do so.

Report

- In addition to the code and agent, you must also submit a report. The report should contain at least the following:
    - a brief overview of the libraries used in your implementation,
    - a complete overview of your model's architecture,
    - a complete overview of the training process, including detailed discussions of any specific techniques and/or algorithms used in your implementation.
    - an explanation of any custom rules that you implemented.

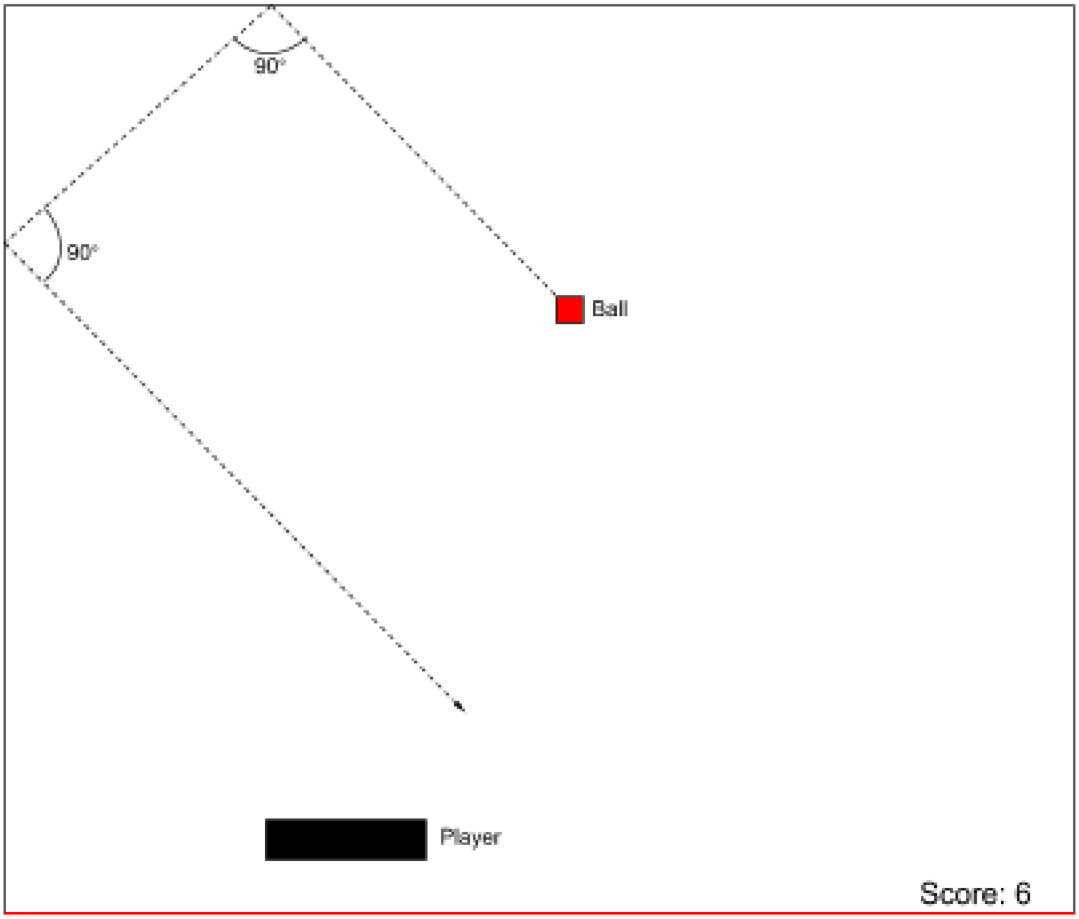
- Where applicable, justify any choices you make in your approach.
- Regardless of which option you choose to implement, also answer the bonus questions (given in appendix A) in your report.

Judging guideline - you will be judged mainly on the merits of your approach. Therefore, a good report will count heavily in your favour, even if the trained model is not perfect.

Option A: Single player Pong

Your goal is to create a game-playing agent that can play single player Pong. The agent should be trained from zero-knowledge in an unsupervised fashion. That is, no game-playing knowledge or strategies may be hard coded, and no training dataset is required.

Example game overview



The game-playing agent controls the black bar at the bottom of the screen. The ball moves at a constant speed, and bounces off the edges of the game area (or the black bar) at a constant angle.

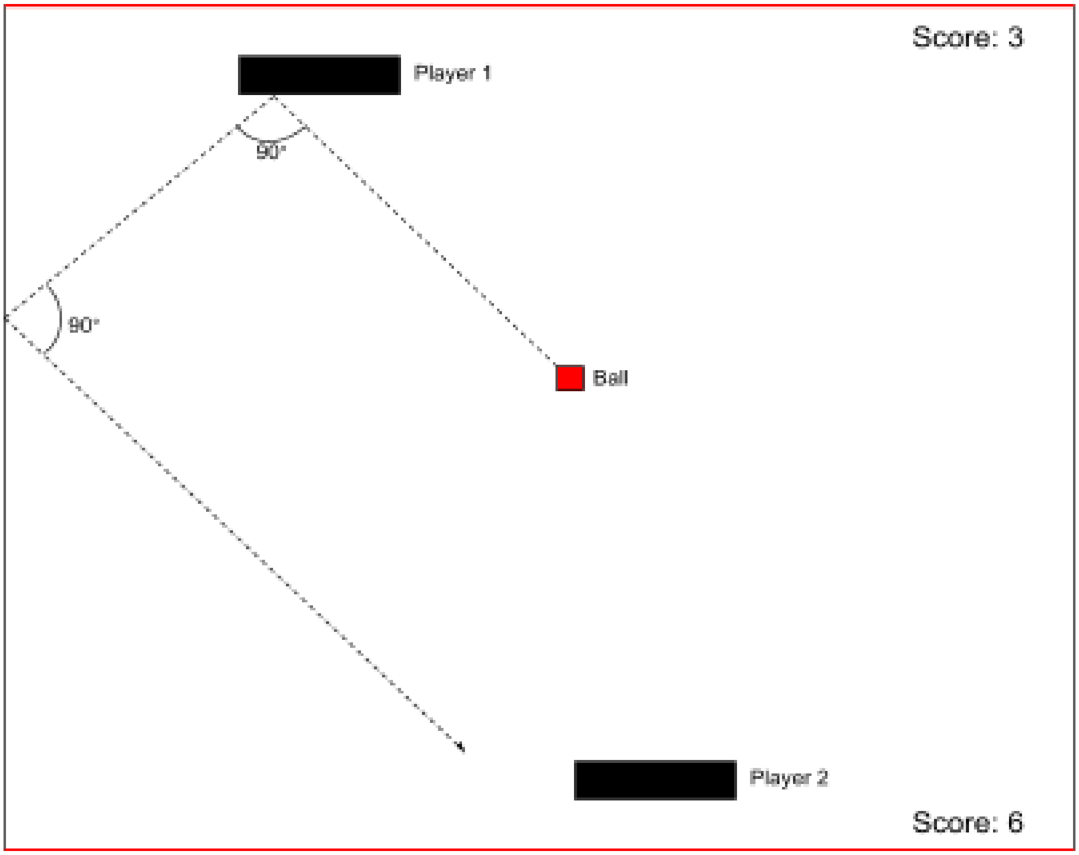
The player’s goal is to position the black bar such that the ball never touches the bottom edge of the game area (shown in red). Whenever the player ‘catches’ the ball, the player gets one point. If the ball ever touches the bottom edge, the player loses and the game ends. You may also impose a maximum score at which point the player wins.

For simplicity, the player’s bar and the ball may move at the same speed. You may choose any size for the game area that you deem appropriate.

Option B: Two player Pong

Your goal is to create two game-playing agents that will compete against each other in two player Pong. The agents should be trained from zero-knowledge in an unsupervised fashion. That is, no game-playing knowledge or strategies may be hard coded, and no training dataset is required. Additionally, the two agents must be trained simultaneously. If you choose this option, please submit *two* trained models (one for each player).

Example game overview



The game-playing agents control the black bars at the top and bottom of the screen. The ball moves at a constant speed, and bounces off the edges of the game area (or the black bars) at a constant angle.

Each player's goal is to position their respective black bar such that the ball never touches the red edge behind their bar. Whenever a player misses the ball, the opposing player gets one point. The first player to get 10 points wins.

For simplicity, the players' bars and the ball may move at the same speed. You may choose any size for the game area that you deem appropriate.

Option B - Bonus Features:

The features described here are not required, but could improve your chances.

1. During training, ensure that each player learns to play on both sides (top and bottom) of the game area.

2. Allow a human player to compete against either of the two trained agents. The human player gets to choose which agent they want to play against, and also whether they want to be the top or bottom player.
3. Gradually increase the speed of the ball as players score points during the game.

Appendix A: Bonus Questions

To improve your chances, answer the following questions in your report. You may include these answers as part of discussions throughout your report, or answer them directly in a separate section.

1. Discuss two machine learning approaches that you did not use in your pong implementation, but that would also be able to perform the task. Explain how these methods could be applied instead of your chosen method.
2. Explain the difference between supervised and unsupervised learning.
3. Explain the difference between classification and regression.
4. In supervised classification tasks, we are often faced with datasets in which one of the classes is much more prevalent than any of the other classes. This condition is called *class imbalance.* Explain the consequences of class imbalance in the context of machine learning.
5. Explain how any negative consequences of the class imbalance problem (explained in question 4) can be mitigated.
6. Provide a short overview of the key differences between deep learning and any non-deep learning method.
7. Make one or two suggestions for how you would implement dynamic difficulty adjustment during gameplay, based on a human player's performance. (This question relates to a bonus features in option B, but does not require that you chose that option, nor that you actually implemented the bonus feature).