Exercise 1: Machine Learning (general) (1.5+1.5=3 Points)

1. Define the terms “supervised learning”, “unsupervised learning”, and “reinforcement learning”.
2. Determine the learning paradigm (supervised, unsupervised, reinforcement) for the following tasks.

*Note: If more than one learning paradigm is possible, select one and provide a brief (1 sentence) explanation.*

(b1) Sentiment analysis (determine if a text has positive or negative sentiment)

(b2) Data compression

(b3) Self-driving cars

(b4) Personalized content recommendation

(b5) Spam filtering

(b6) Sorting fruits in a basket by type

**(a) Definitions**

**Supervised Learning**: You used labeled data in order to train a model, at the end you are able to differentiate or map new inputs that you give it to the model and is going to the obtain correct outputs based on the training data.

**Unsupervised Learning**: The model is trained with unlabeled data, so the model has to identify pattern or characteristics without the guidance of labeled data. Usually, this type or model is useful for clustering and dimensionality reduction.

**Reinforcement Learning**: the model learn to make decisions by taking actions in an environment to maximize cumulative reward, the model receives feedback like “Penalties” or “rewards” based on the actions of the model. he process involves exploration (trying different actions) and exploitation (choosing actions based on learned experience).

**(b) Learning Paradigms**

**(b1) Sentiment analysis**: **Supervised Learning** This task involves labeled data (texts with positive or negative sentiment) to train a model.

**(b2) Data compression**: **Unsupervised Learning** This task identifies patterns in the data without predefined labels, aiming to reduce the size of data while retaining essential information.

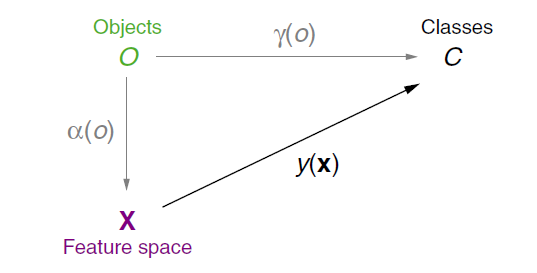
**(b3) Self-driving cars**: **Reinforcement Learning / Supervised Learning** The car learns to navigate and make driving decisions based on feedback from its environment to maximize safety and efficiency, and need a model that has being trained with label data in order to classify in a fast way the images (cars, pedestrians, traffic symbols).

**(b4) Personalized content recommendation**: **Supervised Learning** Explanation: This task often uses labeled data (user preferences) to train a model that predicts what content a user is likely to enjoy.

**(b5) Spam filtering**: **Supervised Learning** Explanation: This task relies on labeled emails (spam or not spam) to train a model to classify incoming messages.

**(b6) Sorting fruits in a basket by type**: **Unsupervised Learning**. This task can involve clustering fruits based on features (like color or shape) without predefined labels.

Exercise 2: From the picture that describes the relationship between real world and model world, the list matches the following symbols



1. A pile of Mushrooms – “**O”** which represent the Objects or Data sets.
2. A table with the columns “size”, “weight”, and “color”, as well as one row for each mushroom, and the respective measurements in the cells – **“X”** which represents or describes the feature space or attributes of each mushroom.
3. A human mushroom expert who can tell whether any mushroom you show them is poisonous or edible – **“Ɣ(O)”** is known as the ideal classifier capable of discerning if a mushroom is edible.
4. A device that measures size, weight and color of a mushroom – **“α(o)”** is known as model formation function that helps to construct the feature space of the attributes of the mushrooms.
5. The set {Poisonous, Edible} – **“C”** is known as the Class or Performance Metric
6. The machine learning system that you are trying to build. – **“y(x)”** is known as the model function which is used for learning edible and poisonous mushrooms.

Exercise 3:

1. See Files in Attachment.
2. I correspond to the **“Ɣ(O)”** symbol which is the ideal classifer . The Symbol **“C”** represents the functions I implemented as I am able to classify human or LLM generated text.

Based on your request, here’s guidance for **Exercise 5** and **Exercise 6** from the uploaded document on rule-based learning:

**Exercise 5: Rule-Based Learning**

**Given Dataset**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

**Step-by-Step Solution**

**(a) Apply the Find-S Algorithm**

• **Find-S** starts with the most specific hypothesis and generalizes it based on positive examples.

Initial hypothesis: h = (?, ?, ?, ?)

• Process each example:

1. Example 1: Positive (Run-a-red-light = yes) → h = (Monday, no, easygoing, evening)

2. Example 2: Negative (Run-a-red-light = no) → No change in h

3. Example 3: Negative (Run-a-red-light = no) → No change in h

4. Example 4: Positive (Run-a-red-light = yes) → Generalize h to match this example:

• h = (Monday, no, easygoing, ?)

Final hypothesis after Find-S: (Monday, no, easygoing, ?)

**(b) Apply the Candidate-Elimination Algorithm**

Candidate-Elimination maintains a version space, represented by the sets **S** (specific boundary) and **G** (general boundary).

**Initialize**:

S = { (Monday, no, easygoing, evening) } (most specific hypothesis)

G = { (?, ?, ?, ?) } (most general hypothesis)

**Process each example**:

1. Example 1: Positive → S remains the same, G remains unchanged.

2. Example 2: Negative → Remove hypotheses in G that match this example but contradict the label. G becomes more specific.

3. Example 3: Negative → Further update G by refining hypotheses that cover this negative example.

4. Example 4: Positive → Adjust S to be less specific if it does not fully match.

**Final Boundary Sets**:

S = { (Monday, no, easygoing, ?) }

G = { (?, ?, easygoing, ?) }

**(c) Version Space**

The version space is the set of hypotheses consistent with all training examples, represented by all hypotheses between S and G.

Final Version Space (hypotheses that satisfy S and G constraints): { (Monday, no, easygoing, ?), (?, ?, easygoing, ?) }

**Exercise 6: Rule-Based Learning (Background)**

**(a) Can a version space contain hypotheses that are neither in the set nor in the set ?**

Yes, a version space can contain hypotheses that are neither in nor . These hypotheses lie between the most specific and most general boundaries and satisfy all positive and negative examples without being the most extreme boundaries.

**(b) Conditions on Hypotheses and in**

For two hypotheses and in (specific boundary):

True, meaning one hypothesis is more general or equal to the other.

False, as both cannot be more general than each other unless they are identical.

**(c) Inductive Bias in Find-S vs. Candidate-Elimination**

**Candidate-Elimination** has a stronger inductive bias compared to **Find-S**. Candidate-Elimination maintains all hypotheses consistent with the data, while Find-S only provides the most specific hypothesis consistent with positive examples. This stronger bias in Candidate-Elimination provides a clearer boundary and multiple hypotheses, giving it a more comprehensive approach to learning.