Exercises

1	2	3	4	5	6

Surname, First name

IFEEMCS520100 Fundamentals of Artificial Intelligence Programme EXAM
Main exam

1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9
0	0	0	0	0	0	0

Student number:

Examiner responsible: dr. Panchamy Krishnakumari

IFEEMCS520100 Fundamentals of AI Programme
17th. Oct 2023, 18:30 - 21:30 (+30min extra time)

This is an examination with **6 sections**. You can score **max 15 points** for a section, and **55 points** in total.
Sections have between 2 and 8 **sub-questions**. For each sub-question, the number of points is given in the **margin** before each question. The division of questions per section is as follows:
Multiple-choice - 4 sub-questions, 5 points
Case study 1 - 6 sub-questions, 15 points
Case study 2 - 2 sub-questions, 5 points
Case study 3 - 3 sub-questions, 7 points
Case study 4 - 5 sub-questions, 15 points
Case study 5 - 4 sub-questions, 8 points
This exam is 24 pages long.

You need to answer all questions **within** the given answer box.
We will scan the exams before grading, and we cannot guarantee that text outside of the answer box will be graded.
We recommend the use of an ink pen, not a graphite pencil, as your answer should be visible after scanning.

This is an **open-book examination**, so the use of books, lecture notes, or other notes is permitted.
You are allowed to use a regular **calculator**.

Check whether you have all the pages and assignments before starting. Each section should take a different amount of time. Do take into account that a sub-question for more points might take more time. Manage your time accordingly.

Multiple-choice

1p **1a**

What will the following code snippet output?

```
df = pd.DataFrame('A': [1, 2, 3], 'B': [4, 5, 6])
print(df.loc[1, 'B'])
```

(a) 1
(b) 5
(c) 6
(d) KeyError

1p **1b** The larger K is in K-means clustering, the better the algorithm typically fits the data.

(a) False (b) True

1p **1c** If you use PCA to project d-dimensional points down to j principal coordinates, and then you run PCA again to project those j-dimensional coordinates down to k principal coordinates, with d > j > k, you always get the same result as if you had just used PCA to project the d-dimensional points directly down to k principle coordinates.

(a) True (b) False

2p **1d** Motivate your choice for question 1c.

The statement is incorrect. Performing PCA in two stages—first reducing the dimensionality from \(d \) to \(j \), then from \(j \) to \(k \)—does not generally yield the same result as directly reducing from \(d \) to \(k \) dimensions. In PCA, each projection is based on the covariance matrix of the data, which captures its variance structure. When you first reduce the data to \(j \) dimensions, some variance is lost. The new \(j \)-dimensional data has a different covariance matrix than the original \(d \)-dimensional data. Running PCA again on this modified data will find principal components based on the new, reduced structure, not the original one. In contrast, directly projecting from \(d \) to \(k \) dimensions selects the top \(k \) principal components based on the full variance of the data. The only case where the two-step process might yield the same result is when the top \(j \) components capture almost all the variance, which is rare. In general, two-step PCA leads to a suboptimal result compared to one-step PCA.
Then when directly reducing from d dimensions to k dimensions, the k eigenvectors selected are the global optimal solution for the entire data set. In the two-step dimensionality reduction process, since the first dimensionality reduction has lost some important variance information, the second dimensionality reduction is a feature decomposition performed on a suboptimal subspace, which cannot guarantee the same result as directly reducing to k dimensions.

Acting humanly	*"The art of creating machines that perform functions that require intelligence when performed by people."* (Kurzweil, 1990) *"The study of how to make computers do things at which, at the moment, people are better"* (Rich and Knight, 1991)
Thinking humanly	*"… to make computers think … machines with minds, in the full and literal sense"* (Haugeland, 1985) *"[The automation of] activities that we associate with human thinking, activities such as decision making, problem solving, …"* (Bellman, 1978)
Thinking rationally	*"The study of mental faculties through the use of computational models."* (Charniak and McDermott, 1985) *"The study of the computations that make it possible to perceive, reason, and act"* (Winston, 1992)
Acting rationally	*"Computational Intelligence is the study of the design of intelligent agents."* (Poole et al., 1998) *"AI … is concerned with intelligent behaviour in artefacts"* (Nilsson, 1998)

Case study 1

When the first chess playing computer beat a human champion, it was considered a great feat of AI. However, nowadays, many people wouldn't call a chess-playing algorithm AI anymore, showing that what is or is not AI is changeable and debated.

2p **2a** Mention one of the four main definitions of AI which would include chess-playing algorithms as AI, and why?

"As an imitation of human behavior": This is a way of defining AI that focuses on AI's ability to imitate specific human behaviors, such as playing chess. Under this definition, chess algorithms are considered part of AI because they are able to imitate and even surpass human performance in chess games. For example, IBM's Deep Blue defeated chess world champion Garry Kasparov in 1997, which was considered a major victory for AI technology because it was able to imitate and optimize human chess strategies.

2b. Definitions of AI that do not include chess algorithms
"As a rational thinker": This definition focuses on AI's decision-making process and problem-solving ability, applied to scenarios that require logic and reasoning. In this view, chess algorithms may not be considered complete AI because they are usually based on preset rules and algorithm optimization, and do not involve true autonomous thinking or the ability to adapt to unknown environments. This definition emphasizes that AI's ability is not just to perform tasks within a specific field, but to more broadly apply its rational thinking to solve a variety of unknown and complex problems.

2p **2b** Mention one of the four main definitions of AI which wouldn't include them, and explain why?

For the following sub-disciplines of AI, discuss in your group if you consider this a goal, or a technique, and what this depends on:

Search
Classification
Emotion recognition
Formal reasoning

3p **2c** In AI subfields, we can distinguish between goals and techniques. For the following, do they describe a goal or technique, and why?

1. Chess
2. Neural networks
3. Winning the imitation game

1. **Chess**
   - **Description**: Chess in the context of AI generally refers to the use of artificial intelligence to play the game of chess against human opponents or other computer programs.
   - **Classification**: **Goal**
   - **Why**: Chess as a domain for AI research is treated as a goal because the primary objective is to develop systems that can successfully understand, strategize, and win at the game of chess. While techniques are used to achieve proficiency in chess (like tree search algorithms, evaluation functions, etc.), "Chess" itself describes the objective or the problem space where these techniques are applied.
2. **Neural Networks**
   - **Description**: Neural networks are computational models inspired by the human brain's structure and function, used widely in AI for tasks such as pattern recognition, classification, and prediction.
   - **Classification**: **Technique**
   - **Why**: Neural networks are a technique because they provide a methodological approach to solving various problems in AI. They are tools or mechanisms employed to achieve certain goals like image recognition, speech translation, or autonomous driving, rather than goals in themselves.
3. **Winning the Imitation Game**
   - **Description**: The imitation game, often referred to as the Turing Test, involves an AI being tested for its ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human.
   - **Classification**: **Goal**
   - **Why**: Winning the imitation game is a goal because it sets a specific objective for AI development—achieving a level of performance in conversational behavior that is indistinguishable from that of humans. The techniques used to achieve this goal could include natural language processing, machine learning models, knowledge representation, etc., but the "winning" aspect itself encapsulates the goal or benchmark that these techniques strive to meet.

Let us look at another great feat of AI - the Paro robot, a small soft robot that looks like a seal. It was designed to bring ease to elderly people with dementia, its purpose was to be a comforting presence. It for instance does this by following voices with its head and moving when touched. For an improved version of Paro a group of engineers wants to use existing speech assistants such as Siri (which is trained on voice interactions from iPhone users) to allow Paro to recognise what people are saying. Paro will not respond with words, but its facial expression and sounds can be adjusted based on the user's speech.

3p **2d** What forms of bias can you identify in this case? Give reasons as to why these are the most likely form(s) of bias

1. Preexisting Bias ↴
Definition: Preexisting bias comes from social institutions, practices, and attitudes, and often exists before the system is created.
Cause: If the speech recognition system used by Paro is trained primarily on data from iPhone users, it may not include speech features of sequences or people with cognitive impairments. Therefore, this bias may arise from the socio-cultural context and social practices of the technology developers, which cover the needs and characteristics of all potential users.
Example: For example, data from iPhone users may be biased towards certain specific geographic regions, economic levels, or age groups, and not fully representative of all sequences, especially those that diagnose dementia.
2. Technical Bias
Definition: Technical bias is caused by technical limitations or considerations of the system itself.
Cause: The algorithm of the speech recognition system may have been originally designed for ordinary adult speech, especially optimized for context or people with dementia. This may cause the algorithm to exhibit bias caused by limitations when dealing with speech technology.
Example: Models trained using standard Mandarin or English may have difficulty accurately recognizing the accent or speech patterns of sequences, especially those that find different speech features due to age-related changes.
3. Emergent Bias
Definition: Cognitive bias generated during use is due to new social knowledge or changes in cultural values, or a mismatch between the users assumed in the system design and the actual user group.
Cause: As Paro increasingly has a diverse background of users, the original design may not meet the needs of all users, especially in terms of cultural and linguistic diversity.
Example: If Paro is mainly used in a specific culture or region, and the reactions and voices used by the robot are designed based on the standards of another culture, it may lead to cognitive cognition, and users may feel that Paro's reactions do not match their cultural expectations.

2p **2e** Identify three value conflicts that are likely to play a role in the case of a Paro that recognises speech and responds with emotions relevant to the situation. Briefly mention the conflicts and suggest trade-off strategies that you consider appropriate to dealing with the different conflicts.

1. Privacy vs. Functionality Conflict

Conflict Description: To enhance the accuracy of Paro's speech recognition and the relevance of its emotional responses, significant amounts of user data, including voice inputs and interaction feedback, may need to be collected and analyzed. This could conflict with users' privacy expectations. **Trade-off Strategy**: Implement a **Satisficing** approach to ensure the system meets an "adequately good" level of privacy protection. This could involve the use of privacy-enhancing technologies such as data anonymization and local processing, while still maintaining the essential functionalities of the system.

2. Personalization vs. General Acceptability Conflict

Conflict Description: For Paro to provide personalized responses, it needs to learn and adapt to specific user behaviors and preferences. However, this personalization could conflict with the goal of designing a system that is universally acceptable and free from biases. **Trade-off Strategy**: Use a **Respecification** approach to rethink how to balance personalization with universality. For example, mechanisms could be designed to allow users to opt-out of certain personalization features, or introduce user feedback loops that enable users to adjust the robot's behavior to better meet their expectations.

3. Safety vs. Natural User Interaction Conflict

Conflict Description: Enhancing Paro's safety to ensure it operates reliably under all circumstances might require limiting some interactive features, which could reduce the naturalness of user interactions. **Trade-off Strategy**: Employ an **Innovation** approach to find new techniques or methods that enhance both the naturalness of interactions and the system's safety. For example, developing new algorithms to improve error detection accuracy or designing more flexible interaction protocols that maintain high safety standards while naturally responding to user needs.

3p **2f** What kind of explanations/information should be available to (1) elderly people using Paro and (2) the nursing staff that cares for these elderly people in order for an enhanced version of Paro to be used in a responsible manner?

1. Explanations/Information for Elderly People Using Paro

For elderly users, particularly those with cognitive or physical limitations, the explanations should be simple, clear, and reassuring. The goal is to foster trust in the system while ensuring they understand how to use it effectively without feeling overwhelmed.

- **Purpose of Paro**: A clear explanation of why Paro is being used (e.g., for comfort, emotional support, or interaction) and how it will respond to their voice or emotions.
- **Basic Interaction Guidelines**: Simple instructions on how to interact with Paro. This includes voice commands it can recognize, what actions or responses to expect, and how Paro interprets their emotional state.
- **Privacy Information**: A simplified explanation of how their data (e.g., voice or behavior patterns) will be used. This should reassure them that their personal information is protected and used only for improving their experience with Paro.
- **Error Handling**: Basic guidance on what to do if Paro does not behave as expected, such as repeating commands or alerting a caregiver if assistance is needed.

2. Explanations/Information for Nursing Staff

For the nursing staff, more detailed information is required since they are responsible for both monitoring Paro's use and ensuring the well-being of the elderly residents. The information should be focused on safety, care integration, and technical support.

- **Technical Functionality**: A comprehensive explanation of how Paro works, including how it recognizes speech, detects emotions, and responds. The staff should understand Paro's capabilities and limitations, such as the types of emotions it can interpret and the actions it can perform.
- **Data Privacy and Security**: Detailed information on how user data (e.g., voice recordings, interaction patterns) is stored, processed, and protected. This should include any legal or regulatory considerations, especially concerning vulnerable populations like the elderly.
- **Monitoring and Oversight**: Guidelines on how to monitor Paro's interactions with the elderly, including how to track whether it is providing appropriate emotional responses or if it is malfunctioning. There should be procedures for reporting or addressing issues if Paro behaves unexpectedly or fails to support the resident effectively.
- **Intervention Protocols**: Clear protocols for when and how the staff should intervene during interactions with Paro. This could include recognizing when Paro's responses might not be appropriate or comforting for the user, or when its usage might exacerbate confusion or frustration in elderly individuals, particularly those with cognitive impairments.
- **Customization and Updates**: Instructions on how to adjust Paro's settings to better suit the individual needs of residents (e.g., adjusting response times, emotional tones, or sensitivity). They should also be informed about software updates, how to install them, and what changes these updates might bring.

Case study 2

2p **3a** Describe briefly the (pool-based) active learning cycle. Explain the least-confidence criteria for data sampling in active learning, and rank the following instances according to the model prediction.

Instance	Class 1	Class 2	Class 3
X1	0.81	0.06	0.13
X2	0.07	0.43	0.5
X3	0.1	0.3	0.6
X4	0.01	0.16	0.83

Pool-based active learning is a form of machine learning where the algorithm iteratively selects the most informative samples from a pool of unlabeled data, queries an oracle (e.g., human annotator) for labels, and updates the model based on the new data. The goal is to maximize the model's performance while minimizing the number of required labeled instances.

Steps in the Pool-Based Active Learning Cycle:

1. **Initial Model Training**: The learning algorithm starts with a small, labeled training dataset and trains an initial model.
2. **Prediction on Unlabeled Data**: The model makes predictions on the pool of unlabeled data.
3. **Query Strategy Application**: A query strategy is applied to select the most informative samples from the unlabeled pool. These samples are where the model is most uncertain thus expected to provide the most value when labeled.
4. **Oracle Labeling**: The selected samples are presented to an oracle (e.g., human expert) to obtain labels.
5. **Model Updating**: The model incorporates these new labeled instances into the training set, and the model is retrained or fine-tuned.
6. **Iteration**: Steps 2 through 5 are repeated until a stopping criterion is met, such as a specific performance level or a maximum number of iterations.

Least-Confidence Criteria for Data Sampling in Active Learning

The **least-confidence sampling** criterion is used to select the data instances about which th current model is least certain regarding its predictions. This criterion typically involves selecting instances where the model's highest predicted probability (confidence) among the classes is the lowest compared to others.

In this sampling method, the instance for which the maximum predicted class probability is the smallest indicates the greatest uncertainty and therefore is considered the most informative for querying the oracle.

Ranking the Instances by Least-Confidence

Given the model predictions for each class of each instance, we rank the instances based on the highest probability (confidence) in descending order of uncertainty (ascending order of confidence):

- For each instance, determine the maximum probability across classes.
- Rank these instances based on these maximum probabilities in ascending order.

Instance Probabilities:

- **X1**: max(0.81, 0.06, 0.13) = 0.81
- **X2**: max(0.07, 0.43, 0.5) = 0.5
- **X3**: max(0.1, 0.3, 0.6) = 0.6
- **X4**: max(0.01, 0.16, 0.83) = 0.83

Ranked by Least Confidence:

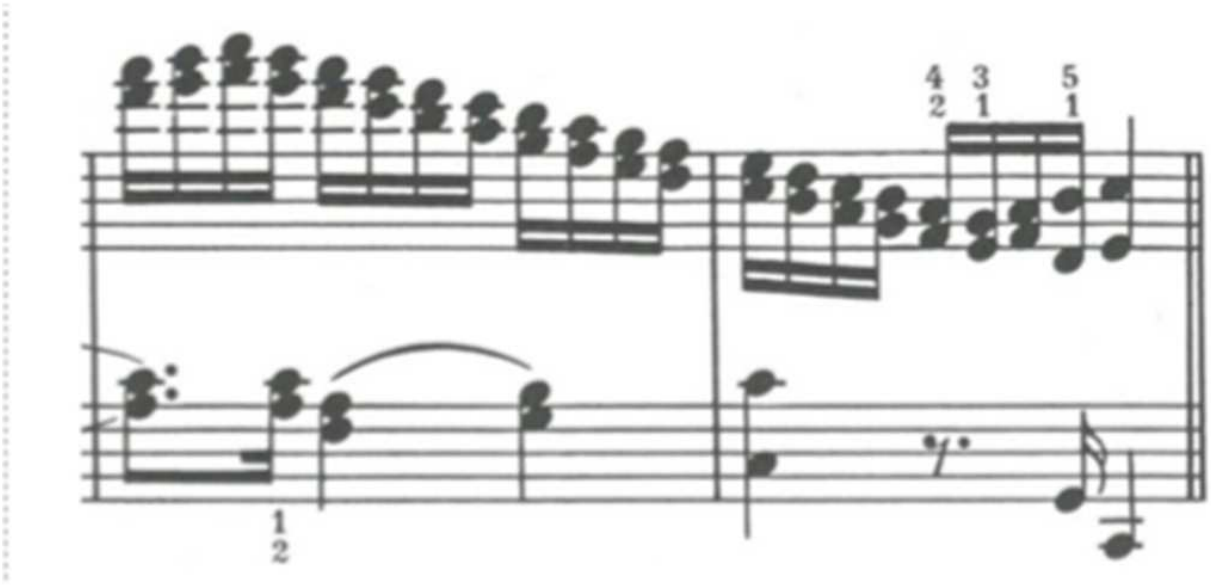
1. **X4** (highest confidence: 0.83)
2. **X1** (highest confidence: 0.81)
3. **X3** (highest confidence: 0.6)
4. **X2** (highest confidence: 0.5)

Conclusion

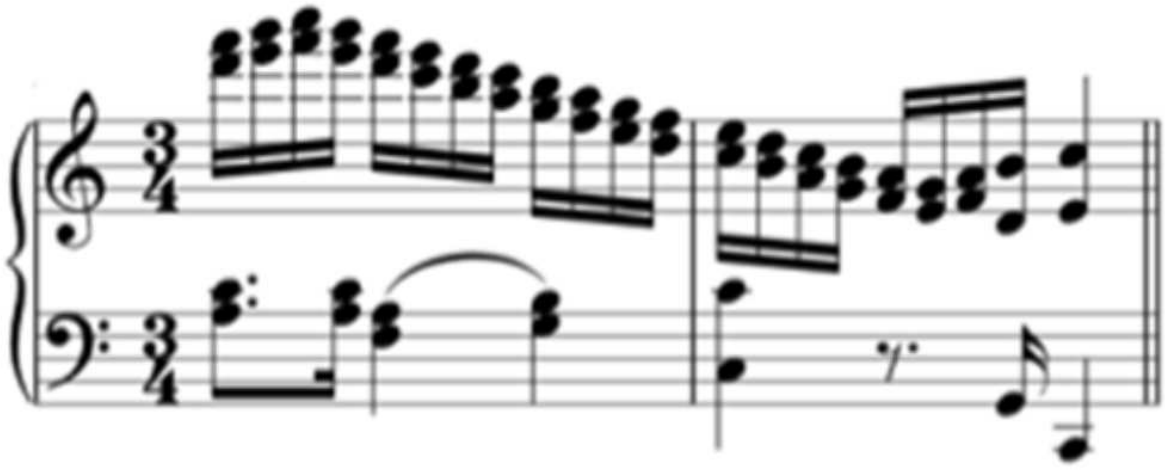
Thus, following the least-confidence criteria, the instance **X2** is the most informative (most uncertain), followed by X3, X1, and X4 in decreasing order of informativeness (increasing order of confidence). This ranking assists in prioritizing which instances to label next to most efficiently improve the model.

3p **3b** You want to build a ML model to help fully digitize musical scores. However as a first step, you want to understand the main errors in digitized musical scores using a particular model. Consider the following task description presented to crowd workers on Amazon Mechanical Turk. Identify 3 potential quality-related issues with this task design. How can they be improved?

In this task, you are required to identify errors in the digitized versions of 25 original musical scores. In each case you will be first presented with the original musical score on a page, and then the digitized score on the next page (see example below). Use the text area below the digitized score to write down the errors you have detected in comparison to the original score. The task is open to all workers on Amazon Mechanical Turk who have completed at least 100 tasks before.



Original Musical Score (displayed on the first page)



Digitized Musical Score (displayed on the second page)

1. Requirement Clarity and Guidance

Issue: The task may be too complex for workers without specific knowledge in reading musical scores, leading to inaccurate error reporting.

Improvement:

- **Enhance Instructions**: Provide more detailed guidelines and examples of common errors to look for in digitized scores, such as misaligned notes, incorrect note values, or missing annotations.
- **Training Module**: Offer a brief training module or tutorial before allowing the workers to start the actual tasks. This could include examples of correct and incorrect digitizations.

2. Qualification of Workers

Issue: While the task requires workers who have completed at least 100 tasks, this does not guarantee they have the necessary expertise in music theory or score reading.

Improvement:

- **Create a Qualification Test**: Develop a specific qualification test that assesses a worker's ability to read and understand musical scores. This test should be passed before allowing them to work on the task.
- **Target Experts**: Consider targeting the task specifically to workers with a background in music or experience in music transcription. This could be identified either through their Mechanical Turk work history or additional screening questions.

3. Comparison Difficulty

Issue: Comparing the original musical score with the digitized version across different pages may lead to errors in detection or reporting due to the difficulty of toggling back and forth.

Improvement:

- **Side-by-Side Display**: Design the interface so that the original and digitized scores can be viewed side-by-side. This would make it easier for workers to spot discrepancies.
- **Interactive Interface**: Implement an interactive tool where workers can mark errors directly on the digitized score in the interface, which can automatically highlight or cross-reference the same elements in the original score.

4. Feedback Mechanism

Issue: Lack of immediate feedback on the accuracy of identified errors can lead to repeated mistakes by workers across tasks.

Improvement:

- **Real-Time Feedback**: Where possible, integrate a system that provides immediate feedback on the worker's performance based on a subset of scores that have been pre-validated. This helps in learning and accuracy improvement.
- **Iterative Review Process**: Allow for an iterative review process where initial findings by one worker are verified by another, adding a layer of quality control.

Case study 3

2p **4a** Explain the concept of Pareto optimality in multi-objective optimization. How does it differ from a single-objective optimization?

Concept of Pareto Optimality

In the context of multi-objective optimization, a solution is considered Pareto optimal if no other solutions in the decision space can improve one objective without causing a deterioration in at least one of the other objectives. These solutions represent the trade-offs among the objectives that cannot be improved upon from one objective's perspective without losing ground on another.

Key Elements of Pareto Optimality:

- **Pareto Dominance**: Solution A Pareto dominates Solution B if A is at least as good as B in all objectives and better in at least one objective.
- **Pareto Front**: The set of all Pareto optimal solutions forms the Pareto front or Pareto boundary. This front is often visualized in the objective space and illustrates the trade-off curve among the objectives.
- **Non-dominated Solutions**: Solutions that lie on the Pareto front are non-dominated, meaning there is no other solution that is better in all the objectives.

Application in Multi-objective Optimization

In practical terms, when dealing with multiple objectives, Pareto optimality provides a framework to guide decision-making:

- **Scenario Evaluation**: In engineering, economics, and management, scenarios can be evaluated based on their placement relative to the Pareto front. Solutions on the front are optimal in the sense that you can't improve any criterion without worsening at least one other.
- **Decision Support**: It supports making decisions where there is no single optimal solution but rather a set of equally valid solutions depending on the decision-maker's preferences or priorities for different objectives.

Comparison with Single-objective Optimization

Single-objective Optimization involves finding the best solution concerning a single criterion. This process is straightforward in terms of decision-making since it involves optimizing one metric, often leading to a unique solution or a set of solutions that can be directly compared through that metric.

Differences from Multi-objective Optimization:

- **Complexity of Solutions**: In single-objective optimization, solutions can be directly compared and ranked. In multi-objective optimization, such direct comparisons are not possible due to the presence of multiple, often conflicting objectives. Instead, solutions are evaluated based on the concept of dominance and trade-offs.
- **Result Interpretation**: Single-objective optimization yields a clear-cut "best" solution, whereas multi-objective optimization results in a set of Pareto optimal solutions, each with different compromises among the objectives.
- **Decision-Making Process**: In single-objective problems, the decision-making process is typically more straightforward since it revolves around maximizing or minimizing a single criterion. In contrast, multi-objective optimization requires a more complex decision-making process where the preferences or weights of different objectives need to be considered, often requiring stakeholder input or sophisticated decision-support tools.

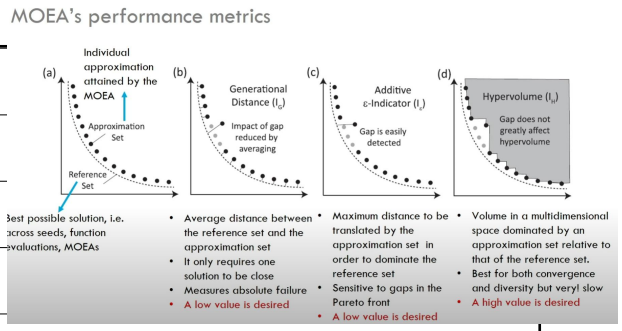
2p **4b** What role does the hypervolume indicator play in evaluating the performance of a multi-objective evolutionary algorithm (MOEA)?

Definition of Hypervolume Indicator

The **hypervolume indicator** (also known as the **S-metric** or **Pareto-compliant indicator**) measures the volume (in the objective space) covered by members of a Pareto front. It is defined as the volume in the objective space enclosed between the Pareto front and a reference point that is typically worse than any feasible solution (e.g., a point that dominates all solutions in the objective space).

Roles of the Hypervolume Indicator

1. **Assessing Convergence**:
   - The hypervolume indicator helps in assessing how close the solutions generated by the MOEA are to the true Pareto front. A larger hypervolume generally indicates that the solutions are closer to the true Pareto front, thus showing better convergence of the algorithm towards optimal solutions.
2. **Measuring Diversity (Spread)**:
   - Besides convergence, diversity among the solutions on the Pareto front is also crucial. The hypervolume indicator naturally incorporates diversity by measuring how broadly the solutions are spread out over the objective space. A higher hypervolume value not only suggests that the solutions are close to the Pareto front but also well-distributed across it.
3. **Comparative Analysis**:
   - Hypervolume provides a single scalar value that summarizes the performance of an MOEA, making it easier to compare the effectiveness of different algorithms or configurations. Algorithms that result in a higher hypervolume are generally considered more effective in solving multi-objective optimization problems.



4. **Selection Pressure**:
   - In MOEAs, the hypervolume indicator can be used directly in the selection process to prefer solutions that contribute most to an increase in the overall hypervolume. This method, known as hypervolume-based selection, can guide the evolutionary process by prioritizing regions of the objective space that are less covered.
5. **Robustness Against Noise and Outliers**:
   - The hypervolume indicator, by considering the aggregate volume covered by the Pareto front, is less susceptible to noise and outliers compared to other metrics that might focus on individual aspects of the solutions.

3p **4c** Formulate a multi-objective optimization problem for a real-world resource allocation scenario. In the context of resource allocation, multi-objective optimization becomes critically important for balancing various competing goals. Your formulation should clearly define the objectives, decision variables, constraints, and any other relevant parameters. For example, you might consider a problem where a city must allocate its annual budget across various sectors like education, healthcare, and public transportation, while optimizing for factors like social welfare, economic growth, and environmental sustainability.

Scenario Overview: A city administration aims to allocate its annual budget across various sectors such as education, healthcare, and public transportation. The goal is to optimize for social welfare, economic growth, and environmental sustainability. These objectives often have competing needs, making it crucial to employ a multi-objective optimization approach.

Decision Variables:

- x_1: Amount of budget allocated to education (in millions of dollars)
- x_2: Amount of budget allocated to healthcare (in millions of dollars)
- x_3: Amount of budget allocated to public transportation (in millions of dollars)

Objectives:

1. **Maximize Social Welfare (SW)**:
   - Social welfare increases with better education and healthcare.
   - $f_1(x) = a \cdot x_1 + b \cdot x_2$
   - Where a and b are coefficients representing the impact of each dollar spent on education and healthcare on social welfare, respectively.
2. **Maximize Economic Growth (EG)**:
   - Economic growth is influenced by all three sectors but most strongly by public transportation improvements which enhance trade and commuting.
   - $f_2(x) = c \cdot x_1 + d \cdot x_2 + e \cdot x_3$
   - Where c, d, and e are coefficients that represent the economic return on investments in education, healthcare, and public transportation, respectively.
3. **Maximize Environmental Sustainability (ES)**:
   - Environmental sustainability is primarily impacted by investments in green public transportation.
   - $f_3(x) = f \cdot x_3$
   - Where f quantifies the environmental benefits per million dollars spent on public transportation.

Constraints:

1. **Budget Constraint**:
   - The total budget allocated must not exceed the total available budget.
   - $x_1 + x_2 + x_3 \leq B$
   - Where B is the total available budget in millions of dollars.
2. **Minimum Sector Investment Constraints**:
   - Each sector has a minimum required investment to maintain basic services.
   - $x_1 \geq M_1$, $x_2 \geq M_2$, $x_3 \geq M_3$
   - Where M_1, M_2, and M_3 are the minimum required investments for education, healthcare, and public transportation, respectively.
3. **Non-Negative Budget Allocation**:
   - The budget allocated to each sector must be non-negative.
   - $x_1, x_2, x_3 \geq 0$

Other Relevant Parameters:

- **Sector Specific Parameters**: These might include the efficiency of fund utilization in each sector, which could adjust the impact coefficients a, b, c, d, e, and f.
- **Policy Restrictions**: Legal or policy constraints might limit the maximum allowable spend in certain areas or dictate certain priorities, influencing the objective functions and constraints.

Case study 4

We are developing a spam filter to classify emails as spam versus non-spam.

2p **5a** Suggest 2 clearly different types of features that could be useful for building a classifier to detect spam.

1. Text-Based Features

Text-based features focus on the content of the email. These features can be derived from the analysis of the words and phrases within the email's body and subject line.

Examples include:

- **Bag of Words (BoW)**: This is a simple yet powerful approach where each word in the email's text is treated as a feature. The value could be binary (indicating the presence or absence of a word) or a count/frequency of the word's occurrence in the email.
- **TF-IDF (Term Frequency-Inverse Document Frequency)**: This is a numerical statistic intended to reflect how important a word is to a document in a collection or corpus. It helps in weighting words differently based on their rarity across all documents (emails), enhancing the importance of more distinctive words.
- **Keyword Identification**: Certain words and phrases are more commonly found in spam emails, such as "free", "guaranteed", "risk-free", and "offer". Identifying the presence and frequency of such keywords can be a strong indicator of spam.

Utility: These features are straightforward to extract and provide a baseline understanding of the textual content, which can be very indicative of spam when certain words or patterns are used excessively or in suspicious contexts.

2. Metadata-Based Features

Metadata-based features derive from the email's header and other non-content aspects th provide contextual and usage-based information. These do not involve the direct textual content of the email body or subject.

Examples include:

- **Sender's Email Address**: The domain or the specific email address of the sender can b indicative of spam. For example, emails from certain domains or those that use numeri or special characters excessively in the username might be more likely to be spam.
- **Time of Sending**: Spam emails might be sent at unusual times compared to typical legitimate communication patterns. Analyzing the time stamps of when emails are sen could provide clues about their legitimacy.
- **Links in the Email**: The number and nature of hyperlinks contained in an email can be indicative of spam. Emails that contain many links, especially if they point to unrecognized or suspicious domains, are more likely to be spam.
- **Attachments**: The presence and types of attachments can be used as features, as spam emails might include executable files or other unusual file types aimed at compromisin security.

Utility: Metadata can often reveal patterns not visible in the content alone and can be particularly useful for identifying mass-spam campaigns or sophisticated phishing attempt that might otherwise linguistically blend in with legitimate emails.

3p **5b** Explain why the Bayes error for spam filtering can be nonzero. Include a definition of Bayes error in your answer.

The **Bayes error rate** represents the lowest possible error rate for any classifier of a random outcome and is analogous to the irreducible error. It is essentially the noise inherent in the underlying problem itself, derived from the probabilistic distribution of the features and labels. The Bayes error occurs because of the natural overlap in the distribution of the classes within the feature space. In mathematical terms, the Bayes error rate is calculated by integrating the minimum of the conditional probabilities of the two classes over the feature space.

Bayes Error in Spam Filtering

In the context of spam filtering, where the task is to classify emails as either spam or non-spam, the Bayes error can be nonzero due to several factors:

1. **Overlap in Feature Distributions**:
   - **Inherent Similarities**: There can be significant overlap in the characteristics (features) of both spam and non-spam emails. For example, both spam and legitimate emails may include similar words such as "offer", "free", or "click here". Even if these words are more frequent in spam emails, their presence in non-spam emails contributes to misclassification risks.
   - **Content Ambiguity**: Certain emails might contain content that is typically associated with both spam and legitimate communications, such as advertisements from reputable companies or promotional emails that users have opted to receive.

2. **Variability and Evolution of Spam**:
   - **Adaptive Spam Techniques**: Spammers continually adapt and modify their content to evade detection, often mimicking the style or format of legitimate emails. This evolution can blur the distinctions used by classifiers to differentiate between spam and non-spam.
   - **Changing Patterns**: As spammers adapt, the probabilistic characteristics of spam can shift, making previously learned patterns less effective and thereby increasing the overlap in class distributions.
3. **Limits of Observable Features**:
   - **Incomplete Information**: The features used in spam filtering (e.g., word frequencies, metadata) might not capture all nuances that distinguish spam from non-spam. Essential cues might be latent or unobservable in the data used for training classifiers.
   - **Contextual and Behavioral Factors**: Factors like the sender's behavior or the recipient's interaction with past emails are difficult to encode purely through email content and metadata but can significantly influence the likelihood of an email being spam.

3p **5c** Explain why for this task the cost of a FP (False Positive) is different from the cost of a FN (False Negative). Which cost do you think is higher? Give an example to illustrate the costs for a user that uses the spam filter and explain what is a FN and a FP in this context.

Definitions in Spam Filtering Context:

- **False Positive (FP)**: This occurs when a legitimate email is incorrectly classified as spam. The email, which is not spam, ends up in the spam folder.
- **False Negative (FN)**: This occurs when a spam email is incorrectly classified as non-spam. The email, which is spam, ends up in the user's inbox.

Cost Differences between FP and FN:

1. **Cost of False Positive (FP)**:
   - **Lost Communication**: A legitimate email marked as spam might contain important information, business opportunities, personal messages, or timely offers. Users might miss crucial communications if they do not regularly check their spam folder.
   - **Business Impact**: For businesses, false positives can mean missing important emails from potential clients or partners, possibly leading to lost revenue or damaged relationships.
   - **User Inconvenience**: Users might have to routinely sift through their spam folder to check for misclassified emails, which is time-consuming and inefficient.
2. **Cost of False Negative (FN)**:
   - **Security Risks**: Allowing spam emails into the inbox can expose users to potentially harmful content, such as phishing scams, malware links, or deceptive content intended to steal personal information.
   - **Reduced Productivity**: Spam emails clutter the inbox, making it harder to manage and sort through important messages, reducing productivity and increasing frustration.
   - **User Annoyance**: Continuous exposure to unwanted emails can be annoying and may lead to a diminished user experience.

Which Cost is Higher?

The relative cost of FPs versus FNs can depend significantly on the context in which the email system is used:

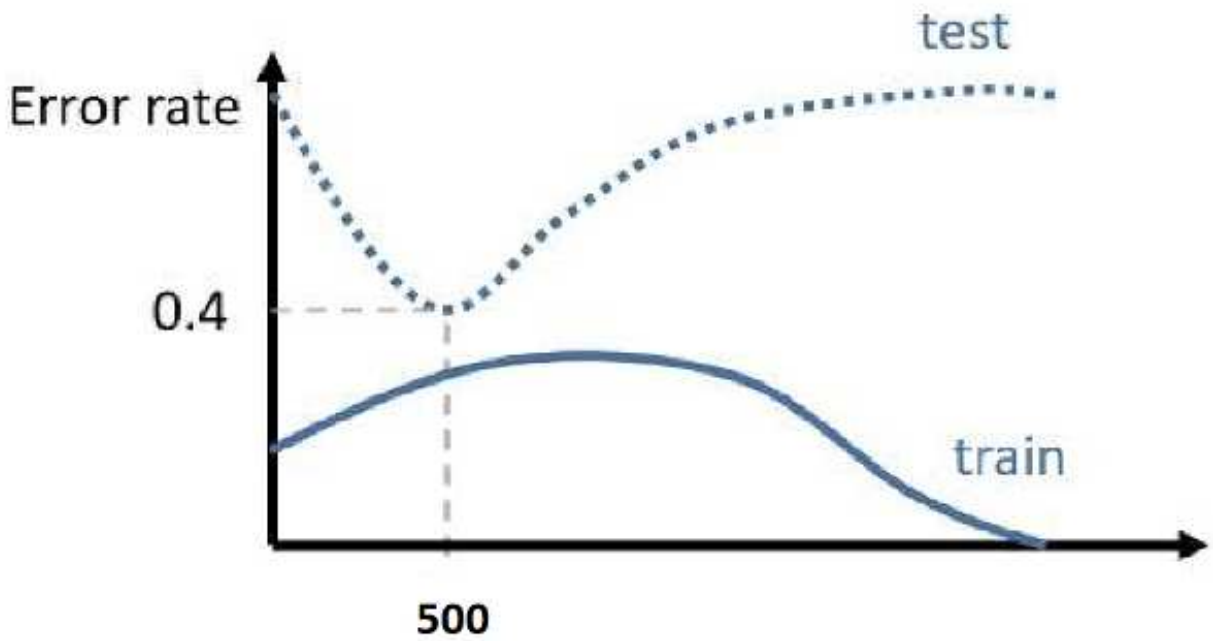
- **For Personal Users**: The cost of false positives might often be considered higher because missing an important personal message (e.g., family communications, job offers, etc.) could have significant personal consequences.
- **For Business Users**: The cost of false negatives may be seen as higher due to the potential for serious security breaches and the disruption caused by spam clogging critical communication channels.

Example Illustration:

Imagine a user who relies on email for business communications:

- **False Positive Scenario**: An email from a new client inquiring about a large order is mistakenly marked as spam. The user does not review their spam folder regularly and misses responding to the email, resulting in a lost business opportunity. Here, the FP h led to a direct financial loss.
- **False Negative Scenario**: A spam email containing a phishing link is incorrectly sent tc the user's inbox. The user clicks the link, leading to unauthorized access to personal data. The FN here results in a security breach and potential financial and reputational damage.

4p **5d** We are using the feature curve (below) to determine that the classifier should use 500 features. Explain why the estimate for the error rate of 0.4 could be too optimistic for the tuned classifier. How should the generalization error for the tuned classifier be estimated instead?



Why the Error Rate Estimate Might Be Too Optimistic

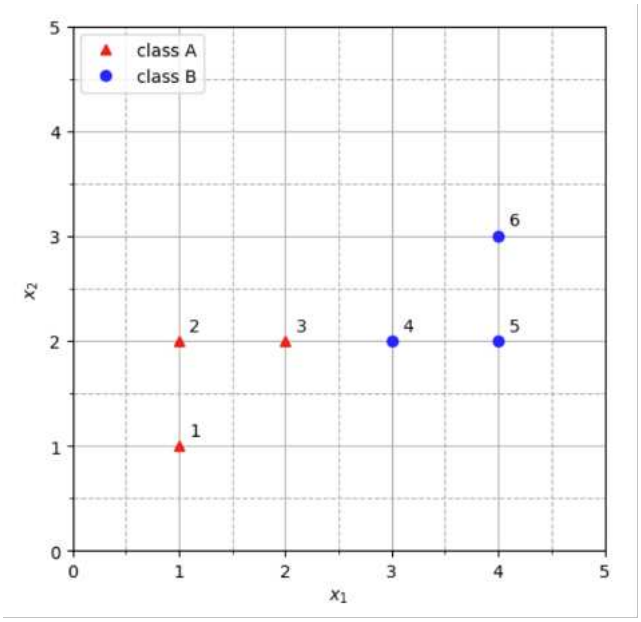
1. **Overfitting at the Chosen Point**: The graph indicates that at around 500 features, there is a significant gap between the training error (which is quite low) and the test error. This gap can be indicative of overfitting—the model performs well on the training data but significantly worse on unseen data. This difference suggests that the model, while optimized for the training set (low training error), may not generalize well outside of this set.
2. **Optimal Feature Count Misinterpretation**: While it appears that using 500 features minimizes the test error, this number of features might not actually offer the best balance between bias and variance. The minimal test error observed in the curve could be a result of particularities in the test set that won't necessarily apply to other unseen data, especially if the test set is not perfectly representative of the general population of data the model will encounter.
3. **Variability in Test Performance**: The test error might also reflect variability in performance that could depend on the specific split of the data into training and test sets. If the dataset split is not robust (e.g., if it's not randomized or if some types of data are overrepresented), the error rate observed could give a skewed view of the true error rate.

Better Estimation of Generalization Error

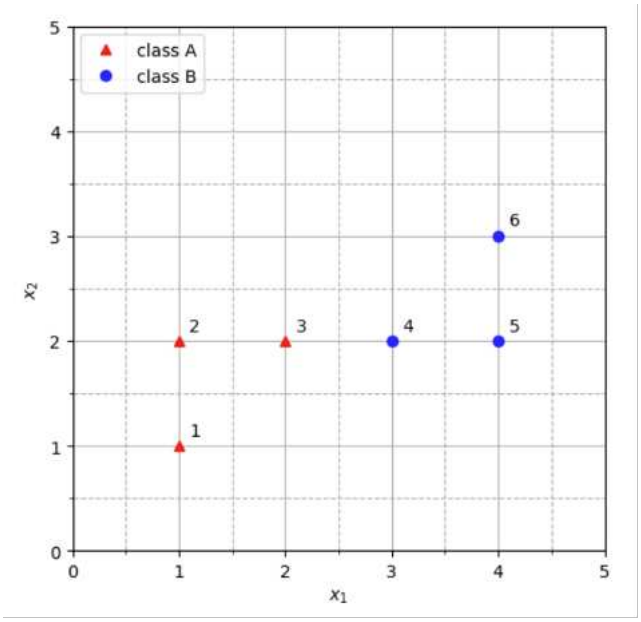
To estimate the generalization error more effectively and realistically, consider the following approaches:

1. **Cross-Validation**: Rather than splitting the dataset into a single training set and test set, use k-fold cross-validation. This technique involves dividing the data into k smaller sets (or folds), using each fold once as a test set while training on the remaining k-1 folds. This process is repeated k times, with each of the k folds used exactly once as the test data. This helps to mitigate the risk that the peculiarities of any single test split unduly influence the error estimate.
2. **Regularization Techniques**: Applying regularization techniques might help in reducing overfitting, particularly when using a large number of features. Techniques such as L1 (lasso) or L2 (ridge) regularization penalize the magnitude of coefficients of features thereby making the model simpler and less likely to overfit.
3. **Feature Selection and Dimensionality Reduction**: Instead of arbitrarily choosing 500 features based on where the test error appears minimized, use systematic feature selection techniques or dimensionality reduction methods. Techniques such as forward selection, backward elimination, or using principal component analysis (PCA) could help in selecting the most relevant features that contribute to model performance without overfitting.
4. **Ensemble Methods**: Using ensemble methods like bagging or boosting can help in improving the generalization ability of the model. These methods combine the predictions of several base estimators to improve robustness and balance out errors.
5. **Analyzing Learning Curves**: Further analyze the learning curves by increasing the dataset size and observing the behavior of the training and test errors. If the gap between them narrows with more data, it indicates that gathering more data could be a solution to achieve better generalization.

3p **5e** Using the 2 most informative features and a small subset of the spam database we obtain the following scatterplot (below).

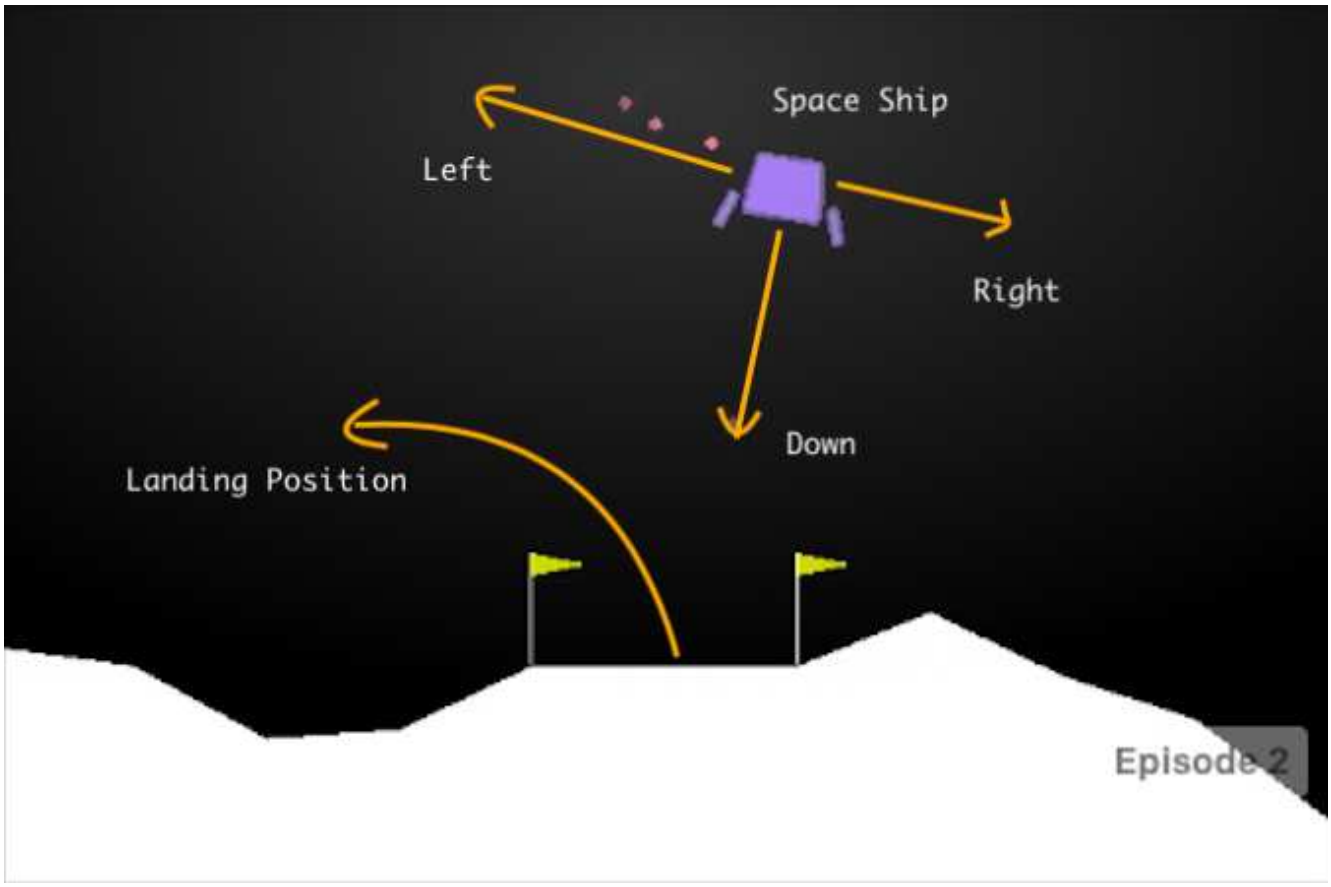


Assume we train a hard-margin support vector machine classifier on this data. Draw the decision boundary, the margin, and indicate the support vectors in the figure below:



Case study 5

The picture below shows the 2D "MoonLander" environment. Here the space ship has to be landed (come to rest) in the landing position between the two flags without crashing (contact with some velocity) anywhere onto the moon's surface. The ship has 4 discrete actions: it can change it's angular velocity by firing the left (a_1) or right (a_2) thruster, accelerate the ship by firing the main downwards engine (a_3) or do nothing (a_4).



3p **6a** Define a low-dimensional continuous state space for the MoonLander

1. **Position (x, y):** These coordinates determine the current location of the space ship in the 2D environment. The x-coordinate can represent the horizontal position relative to the starting point, and the y-coordinate can represent the vertical position or altitude from the surface.
2. **Velocity (vx, vy):** The horizontal (vx) and vertical (vy) velocities are crucial for determining the current motion state of the space ship. These velocities will help predict future positions and are essential for controlling the landing.
3. **Angular Position (θ):** The orientation of the space ship, which affects its descent dynamics and will be crucial when applying thrusts for corrections.
4. **Angular Velocity (ω):** This represents the rate of change of the angular position, indicating how fast the space ship is rotating. This helps in predicting future orientations and is necessary for stabilizing the ship.

1p **6b** Define a low-dimensional continuous action space the MoonLander could use

3p **6c** Define a reward function, which would incentivize a reinforcement learning algorithm to learn the above task with minimal amounts of expended thrust and time. Explain why your reward solves the task.

1. **Successful Landing Reward**: Give a substantial positive reward when the lander successfully touches down at the designated landing area between the flags without crashing and at a safe speed. This component ensures that the primary goal is to achieve a safe and precise landing.
2. **Crash Penalty**: Impose a significant negative reward if the lander crashes or lands outside the designated area. This penalty discourages unsafe maneuvers and incentivizes precision.
3. **Fuel Efficiency Incentive**: Deduct points based on the amount of fuel used during the descent. This could be formulated as a negative reward proportional to the thrust output, thereby encouraging conservation of fuel.
4. **Time Efficiency Incentive**: Apply a small penalty for each timestep taken to reach the ground to encourage quick descents without compromising safety.

1p **6d** Which deep reinforcement learning algorithm from the lecture could be used to learn this task? You do not need to justify your answer.

Deep Q-Networks (DQN) could be used to learn this task.

This page is left blank intentionally