# ICT304 Tutorial 4

Table of Contents

[*1. AI System Engineering: Key Challenges and Lessons Learned* 1](#_Toc178969042)

[*1.1 Introduction* 1](#_Toc178969043)

[*1.2 Key Challenges* 1](#_Toc178969044)

[*1.3 Lessons Learned* 1](#_Toc178969045)

[*1.4 Considerations for Future AI Initiatives* 2](#_Toc178969046)

[*2. Responsible AI* 3](#_Toc178969047)

[*2.1 Overview* 3](#_Toc178969048)

[*2.2 Core Principles* 3](#_Toc178969049)

[*2.3 Significance in AI System Design* 3](#_Toc178969050)

[*3. Evaluation of Classifier Performance* 4](#_Toc178969051)

[*3.1 Confusion Matrix Overview* 4](#_Toc178969052)

[*3.2 Performance Metrics* 5](#_Toc178969053)

[*3.3 Summary of Performance Metrics* 5](#_Toc178969054)

[*4. Edge Cases vs. Outliers* 6](#_Toc178969055)

[*4.1 Edge Cases* 6](#_Toc178969056)

[*4.2 Outliers* 7](#_Toc178969057)

[*4.3 Key Differences* 8](#_Toc178969058)

## *1. AI System Engineering: Key Challenges and Lessons Learned*

### *1.1 Introduction*

AI system engineering encompasses a range of intricate technical and organisational obstacles that must be addressed to develop dependable and scalable AI solutions. This paper outlines significant challenges encountered in the domain and offers valuable insights from various AI initiatives.

### *1.2 Key Challenges*

1. **Data Management:** It is crucial to maintain high standards of data quality, diversity, volume, and privacy.
2. **Model Integration:** Integrating machine learning models into large systems can be complex and requires careful planning.
3. **Performance Monitoring:** Ongoing tracking is essential to detect model drift and ensure stable performance.
4. **Explainability:** Ensuring transparency in AI systems is vital for fostering trust, particularly in critical areas such as healthcare.
5. **Ethics and Regulation:** It is essential to align AI systems with ethical principles and regulatory requirements to minimise risks.

### *1.3 Lessons Learned*

1. **Iterative Development:** An iterative approach allows greater flexibility in addressing changing requirements and system complexities.
2. **Collaboration:** Interdisciplinary teamwork contributes to the robustness of AI systems.
3. **Operationalisation:** Continuous Integration / Continuous Deployment (CI/CD) pipelines facilitate efficient deployment and ongoing maintenance.
4. **Post-Deployment Monitoring:** Regular feedback and monitoring are essential for preserving system integrity and performance over time.

### *1.4 Considerations for Future AI Initiatives*

Implementing the insights from this paper is crucial for future AI projects, such as improving the reliability of AI-driven security systems. Emphasising iterative methods, prioritising transparency, and establishing comprehensive monitoring protocols can enhance the trustworthiness and adaptability of these systems.

## *2. Responsible AI*

### *2.1 Overview*

Responsible AI is the practice of developing AI systems that prioritise ethical, legal, and social considerations. It emphasises building AI technologies that operate fairly, transparently, and securely while safeguarding user rights and addressing potential societal impacts.

### *2.2 Core Principles*

* **Fairness:** Addressing bias and guaranteeing equitable treatment for all groups.
* **Transparency:** Ensuring that AI decision-making processes are comprehensible to users.
* **Accountability:** Defining clear responsibility for the outcomes generated by AI systems.
* **Privacy:** Protecting personal information from unauthorised access.
* **Safety:** Avoiding harm and ensuring robust security measures within AI systems.

### *2.3 Significance in AI System Design*

Emphasising responsible AI contributes to creating trustworthy systems, ensuring adherence to ethical and regulatory standards, and minimising the risk of unintended negative impacts. Incorporating these principles in AI development results in safer, more dependable, and socially advantageous systems.

## *3. Evaluation of Classifier Performance*

### *3.1 Confusion Matrix Overview*

The classifier’s performance on a testing set is summarised in the confusion matrix below:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive (pos)** | **Predicted Negative (neg)** |
| **True Positive (pos)** | 50 | 50 |
| **True Negative (neg)** | 40 | 850 |

From the confusion matrix, the fundamental quantities are extracted:

* True Positives (TP) = 50
* False Positives (FP) = 40
* True Negatives (TN) = 850
* False Negatives (FN) = 50

### *3.2 Performance Metrics*

**Precision:**

Precision measures the proportion of correct identifications. It is calculated as:

**Recall (Sensitivity):**

Recall, also known as Sensitivity, evaluates the proportion of actual positives that were correctly identified by the classifier:

**Specificity:**

Specificity measures the proportion of actual negatives that were correctly identified by the classifier:

**G-Mean:**

The G-Mean is the geometric mean of Recall and Specificity. It provides a balanced measure of classifier performance for both positive and negative classes:

### *3.3 Summary of Performance Metrics*

The calculated values for the classifier’s performance metrics are summarised below:

* **Precision:** 0.5556
* **Recall (Sensitivity):** 0.5
* **Specificity:** 0.9551
* **GMean:** 0.6911

## *4. Edge Cases vs. Outliers*

In machine learning and software development, edge cases and outliers signify exceptional situations that may test a model or system's overall performance or behaviours. While these terms are often used interchangeably, they represent different concepts with unique characteristics.

### *4.1 Edge Cases*

**Definition:**

An edge case refers to a scenario occurring at the extreme limits of the inputs or conditions a system is designed to manage. These situations, while race, are plausible within the expected range of input data. Edge cases typically challenge a system’s robustness, as they exist at the “edges” of its operational capabilities.

**Example:**

Consider a facial recognition system intended to identify human faces in images. An edge case might arise when the input image features several faces partially hidden by shadows or objects. Although the system is built to work with images containing faces, this specific situation, while still within the domain of face detection, poses an unusual challenge due to the obstructions present.

**Key Characteristics:**

* Occurs at the boundaries or limits of standard inputs.
* Expected and falls within the design parameters of the system.
* Often tests the system’s robustness and ability to manage extreme conditions.

### *4.2 Outliers*

**Definition:**

Outliers are data points that significantly differ from the majority of the dataset. These are unexpected stances that fall outside the normal distribution of data and may arise from rare occurrences, data inaccuracies, or noise. While edge cases represent legitimate inputs, outliers typically signify anomalies that are not anticipated within the normal functioning of a model.

**Example:**

In a dataset detailing house prices in a suburban neighbourhood, most properties are valued between $200,000 and $500,000. However, a data point reflecting a house priced at $10 million would be classified as an outlier because it is far removed from the typical range for that area. This price could result from a unique luxury property or a data entry mistake that deviates from the expected distribution.

**Key Characteristics:**

* Unexpected and diverged from the norm.
* Usually lie outside the range of standard inputs.
* May indicate data anomalies, noise, or rare occurrences.

### *4.3 Key Differences*

**Nature of the Data:**

Edge cases are within expected inputs but represent extreme or rare scenarios. In contrast, outliers are atypical data points that lie beyond the anticipated range and often signal something abnormal.

**Impact on System Design:**

Edge cases generally assess how effectively a system can manage extreme yet valid inputs. Conversely, outliers may highlight errors or noise in the data and can significantly affect model accuracy, potentially leading to misclassification if not appropriately addressed.

**Handling in Models:**

Systems are frequently designed to accommodate edge cases, ensuring robust performance even at operational limits. Outliers, however, may be removed or down-weighted during data preprocessing to enhance model performance, as they can skew statistical measures or disrupt model learning.