AI in Society and Public Services

Session 3: The Lifecycle of a Machine Learning Model

Mário Antunes

January 28, 2026

Universidade de Aveiro

Table of Contents i

AI in Society and Public Services

Part 1: From Problem to Model

Part 2: Training & Evaluation

Part 3: The Importance of Retraining

Conclusion

Q&A

AI in Society and Public Services

Session details i

Session 3: The Evolution of Machine Learning Models

From Simple Algorithms to Transformers

Duration: 3 Hours

Instructor: Mário Antunes

Session details ii

Scan the QR code below to access all slides, code examples, and resources for this workshop.

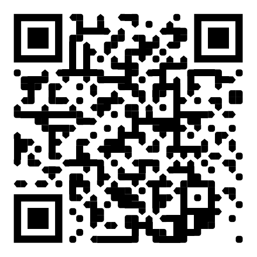

Figure 1: Repository QR Code

Link: https://github.com/mario-antunes/aiml-society

Part 1: From Problem to Model

1.1 The Reality of ML Projects

The 80/20 Rule in Data Science: While academia often focuses on model architecture (the 20%), industry reality dictates that **80% of the effort** lies in data preparation and problem definition.

The ML Lifecycle Stages:

1. **Business Understanding:** Mapping abstract needs to mathematical problems.
2. **Data Acquisition & Understanding:** The raw material.
3. **Data Preparation:** The refining process (ETL).
4. **Modeling:** The algorithmic core (Discriminative or Generative).
5. **Evaluation:** Statistical and Business validation.
6. **Deployment:** Integration into production agents.

1.2 Defining the Business Problem i

Case Study: Predicting Public Service Absenteeism

- **The Symptom:** “We don’t have enough staff on Mondays.”
- **The Analytical Question:** Can we estimate the probability $P(y|X)$ where y is ‘Absent’ and X are employee/environmental features?
- **Target Variable:** Binary (0=Present, 1=Absent) or Continuous (Hours absent)? *Let’s assume Binary Classification.*

1.2 Defining the Business Problem ii

Crucial Feasibility Check:

- Is there a pattern? (Randomness cannot be predicted).
- Do we have data? (History).
- **Is ML necessary?** If a rule `IF (Flu_Season AND Monday) THEN (High_Risk)` works, do not use a Neural Network.

1.3 Data Collection & Preparation

Data is rarely “Model-Ready”.

- **Data Cleaning (Sanitization):**
  - *Missing Values:* Imputation (Mean/Median) vs. Dropping rows. *Warning: Imputing with mean reduces variance.*
  - *Outliers:* Is an age of 120 an error or a super-centenarian? (Z-Score analysis).
- **Formatting:**
  - Date parsing (`2023-01-01` → Timestamp).
  - Text encoding (UTF-8 corrections).

1.4 Feature Engineering: The Art of ML

Feature Engineering is the process of using domain knowledge to extract features that make ML algorithms work better.

1. **Categorical Encoding:**
   - *One-Hot Encoding:* `Department: HR` → `[0, 0, 1, 0]`. (Safe for non-ordinal data).
   - *Label Encoding:* `Low, Medium, High` → `1, 2, 3`. (Preserves order).
2. **Temporal Features:**
   - Raw Date is useless.
   - *Engineered:* `Is_Monday?`, `Distance_to_Payday`, `Season`.
3. **Normalization/Scaling:**
   - Algorithms like KNN or Gradient Descent require features on the same scale (e.g., 0-1) to converge efficiently.

Part 2: Training & Evaluation

2.1 The Training Loop: How Machines “Learn” i

Learning is an **Optimization Problem**. We aim to minimize a Loss Function $J(\theta)$.

The Cycle (Epochs):

1. **Forward Propagation:** Input X passes through weights W and biases b.

$$\hat{y} = \sigma(WX + b)$$

2. **Loss Calculation:** Compare Prediction $\hat{y}$ vs. Ground Truth y.
    - *MSE* (Regression) or *Cross-Entropy* (Classification).

2.1 The Training Loop: How Machines “Learn” ii

3. **Backward Propagation:** Calculate the gradient of the loss with respect to weights.

$$\nabla_\theta J(\theta)$$

4. **Optimizer Step (Gradient Descent):** Update weights to reduce error.

$$\theta_{new} = \theta_{old} - \alpha \cdot \nabla J$$

(α = Learning Rate)

2.2 Evaluation Metrics: Beyond Accuracy

In our Absenteeism case, classes are **Imbalanced** (e.g., 95% Present, 5% Absent).

- **Accuracy Paradox:** A model that predicts “Always Present” has 95% Accuracy but is useless.

The Confusion Matrix:

- **True Positive (TP):** Correctly predicted Absent.
- **False Positive (FP):** Predicted Absent, but was Present (False Alarm).
- **False Negative (FN):** Predicted Present, but was Absent (Missed Detection).
- **True Negative (TN):** Correctly predicted Present.

2.3 Precision, Recall, and F1 i

- **Precision (Quality):** “When it rings, can I trust it?”

  $$\frac{TP}{TP + FP}$$

  High Importance if intervention is costly (e.g., firing someone).

- **Recall (Quantity):** “Did we catch them all?”

  $$\frac{TP}{TP + FN}$$

  High Importance in our case: We need to cover shifts. A False Alarm is better than a no-show.

2.3 Precision, Recall, and F1 ii

- **F1-Score:** Harmonic mean (punishes extreme values).

$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

2.4 Overfitting vs. Underfitting

The fundamental trade-off in ML: **Bias vs. Variance.**

- **Underfitting (High Bias):** Model is too simple to capture the pattern. (e.g., trying to fit a curve with a straight line).
  - *Fix:* Increase model complexity, add features.
- **Overfitting (High Variance):** Model memorizes the noise in the training set.
  - *Symptom:* Training Accuracy »> Validation Accuracy.
  - *Fix:* Regularization (L1/L2), Dropouts, Early Stopping, Cross-Validation (K-Fold).

2.5 Generative AI: A Paradigm Shift

Moving from **Discriminative AI** (Classifying $X \to Y$) to **Generative AI** (Modeling $P(X)$ to create new X').

The Core Concept: Latent Space

- High-dimensional data (images/text) is compressed into a lower-dimensional “Latent Space” (manifold).
- In this space, semantically similar items are topologically close.
- **Generation** = Sampling a point in this latent space and decoding it back to pixels/text.

2.6 Large Language Models (LLMs)

Architecture: Transformer (Decoder-only usually).
Mechanism: Autoregressive Next-Token Prediction.

1. **Tokenization:** Text $\longrightarrow$ Integers.
2. **Embedding:** Integers $\longrightarrow$ Dense Vectors (Latent Space).
3. **Attention Layers:** The model weighs the relationship between every token (Context).
4. **Probabilistic Output:**

   $$P(w_t | w_{t-1}, w_{t-2}, ...)$$

   The model outputs a probability distribution over the entire vocabulary.

Latent Space Dynamics: LLMs traverse the latent space of "meanings" step-by-step. The path is determined by the *Temperature* (randomness).

2.7 Stable Diffusion (Image Generation)

Architecture: U-Net + Variational Autoencoder (VAE) + CLIP.
Mechanism: Iterative Denoising.

1. **Training:** Take an image, add Gaussian noise until it is pure static. Teach the neural network to predict the noise that was added.
2. **Generation:** Start with random static (Latent noise). Ask the model: “What noise would I remove to make this look like a ‘Cat’?”
3. **Iterative Refinement:** Repeat this process 20-50 times.

2.8 Comparison: Text vs. Image GenAI

Feature	LLM (Text)	Diffusion (Image)
Fundamental Unit	Discrete Tokens (Categorical)	Continuous Variables (Pixel/Latent values)
Generation Flow	**Linear/Serial:** $A \to B \to C$. The past dictates the future.	**Refinement/Parallel:** Coarse $\to$ Fine. The whole image emerges at once.
Latent Space	**Semantic:** “King” near “Queen”. Structure is syntactic.	**Visual/Spatial:** “Texture” and “Shape” clusters. Continuous interpolation.
Error Modes	**Hallucination:** plausible but false facts.	**Artifacts:** Extra fingers, impossible geometry.

2.9 Explainable AI (XAI) i

The “Black Box” Problem: Deep Learning models (especially GenAI) operate with millions/billions of parameters. We cannot trace the logic manually.

2.9 Explainable AI (XAI) ii

Why XAI is mandatory:

1. **Debugging:** Is the model looking at the “Absenteeism” pattern, or just the fact that the file name starts with “A”? (Clever Hans effect).
2. **Fairness:** Ensuring decisions aren’t based on protected attributes (Gender, Race).
3. **Trust:** Stakeholders will not use an autonomous agent they do not understand.

2.10 XAI Techniques

Global Explanation (How the model works overall):

- *Feature Importance:* “Tenure” is the most predictive variable globally.

Local Explanation (Why it made THIS decision):

- **LIME / SHAP:** Perturb the input and see how the prediction changes.
  - *Example:* “This employee was predicted ‘Absent’ because ‘Distance > 50km’ pushed the probability up by 15%.”
- **Saliency Maps (Images):** Heatmaps showing which pixels triggered the neuron.
- **Attention Weights (LLMs):** Visualizing which previous words the model focused on when generating the current word.

Part 3: The Importance of Retraining

3.1 Models are Perishable Goods

Unlike standard software code (which doesn’t rot), **ML models degrade**. The world changes, but the model’s weights remain frozen at the time of training.

Model Drift: The divergence between the model’s training environment and the production environment.

3.2 Types of Drift

1. **Data Drift (Covariate Shift):**
   - The distribution of input variables (X) changes.
   - *Example:* A hiring freeze means the average “Tenure” of employees increases significantly. The model has never seen such high tenure values.
2. **Concept Drift:**
   - The relationship between Input and Target ($X \to y$) changes.
   - *Example:* Before COVID, “Working from Home” meant “Sick”. After COVID, “Working from Home” is normal. The *meaning* of the feature changed.

3.3 The Maintenance Cycle (MLOps)

To maintain an Autonomous Agent, we need an automated pipeline:

1. **Monitoring:** Dashboard tracking distribution of inputs (Kullback-Leibler divergence) and output accuracy.
2. **Retraining Triggers:**
   - *Time-based:* Every month.
   - *Performance-based:* When F1-score drops < 0.7.
   - *Data-based:* When drift is detected.
3. **The Feedback Loop:**
   - New data must be captured, labeled (Ground Truth), and fed back into the training set.

Conclusion

Summary of Session 3

- **Problem Definition:** Don't start coding until the business value and target metrics (Precision vs Recall) are defined.
- **Data:** Quality > Quantity. Feature Engineering drives performance.
- **GenAI:**
    - LLMs predict **next tokens** (Sequence).
    - Diffusion predicts **noise** to subtract (Refinement).
    - Both rely on manipulating vectors in **Latent Space**.
- **XAI:** Essential for ethical and robust agents.
- **Lifecycle:** Deployment is just the beginning. Monitor for **Drift**.

Q&A