FUNDAMENTALS OF MACHINE LEARNING FOR ADDITIVE MANUFACTURING

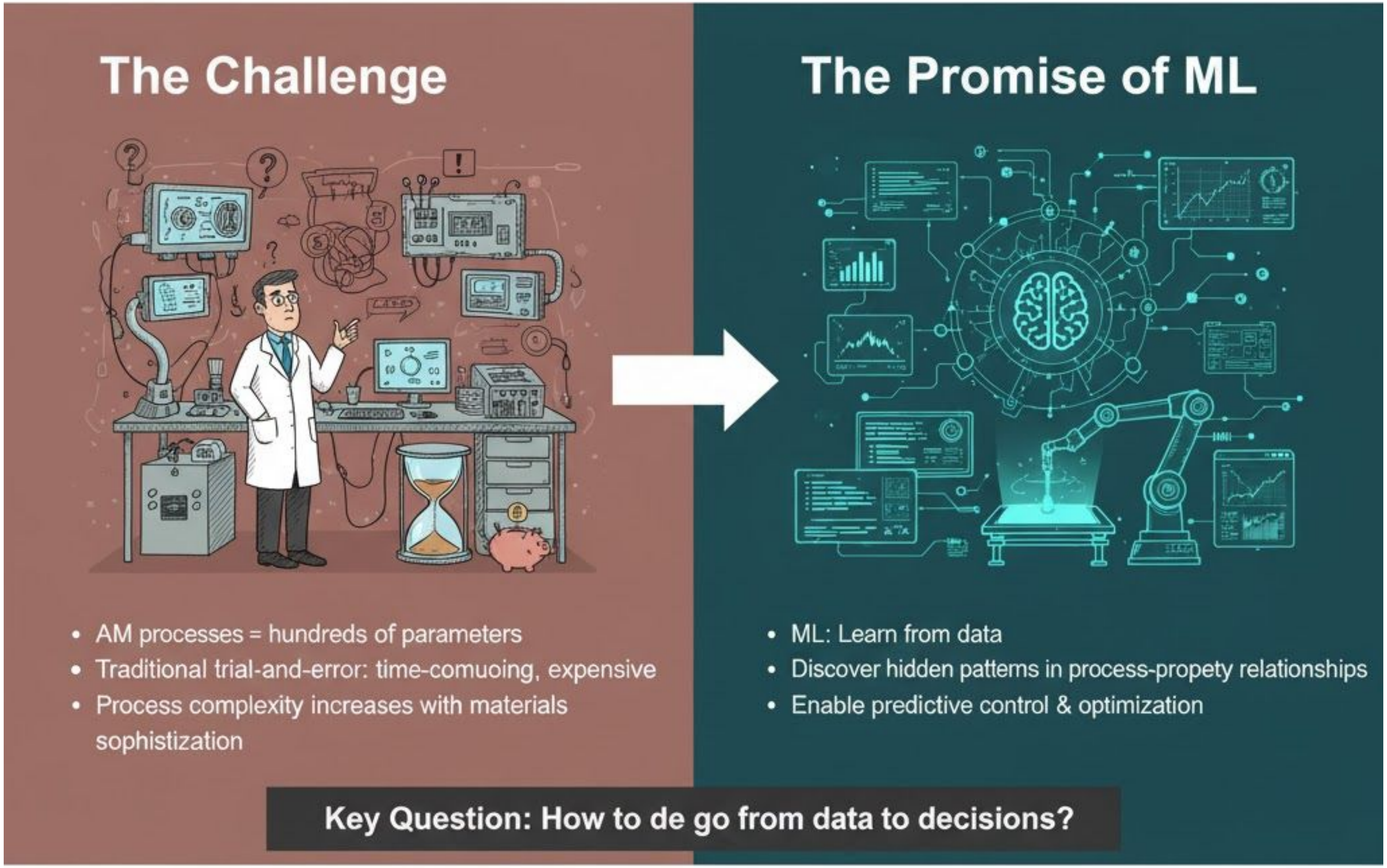
Lecture 1 of 3

Dr. Nazarii Mediukh,
Institute for Problems of Materials Science, NASU

Supported by

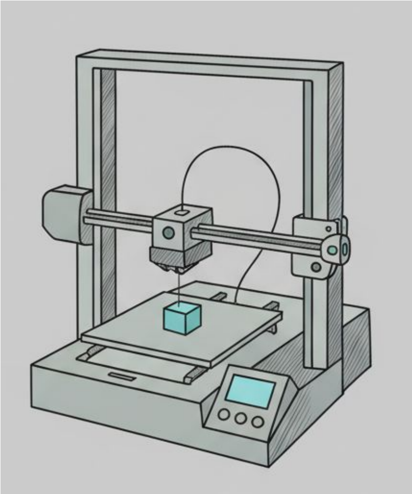
Co-funded by the European Union

What is AI/ML and why do we need it?

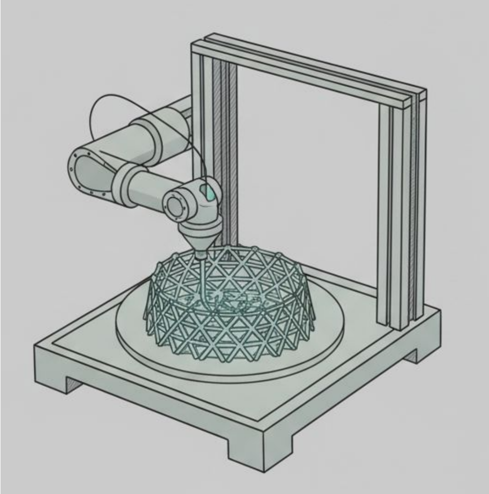


The AM complexity spectrum

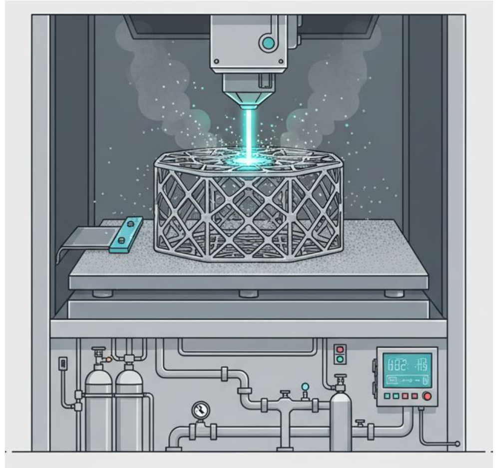
Simple processes (FDM)
- ~5-10 primary parameters (temperature, speed, layer height)
- Relatively predictable behavior
- Desktop accessibility



Complex processes (Robocasting, Binder jettings)
- 15-25+ parameters (rheology, pressure, nozzle geometry, drying)
- Material-dependent behavior (viscosity, yield stress, thixotropy)
- Time-dependent phenomena (solvent evaporation, curing)



Highly Complex (Metal AM)
- 30+ parameters (laser power, scan speed, hatch spacing, atmosphere)
- Multi-physics phenomena (thermal, mechanical, fluid dynamics)
- Defect formation mechanisms (porosity, cracking, warping)
- Insight: Complexity drives the need for data-driven approaches



Course roadmap

Lecture 1: Foundations (Today)
- ML basics
- Data in AM
- The ML pipeline

Lecture 2: Process optimization
- Regression techniques
- Bayesian optimization
- Multi-fidelity approaches

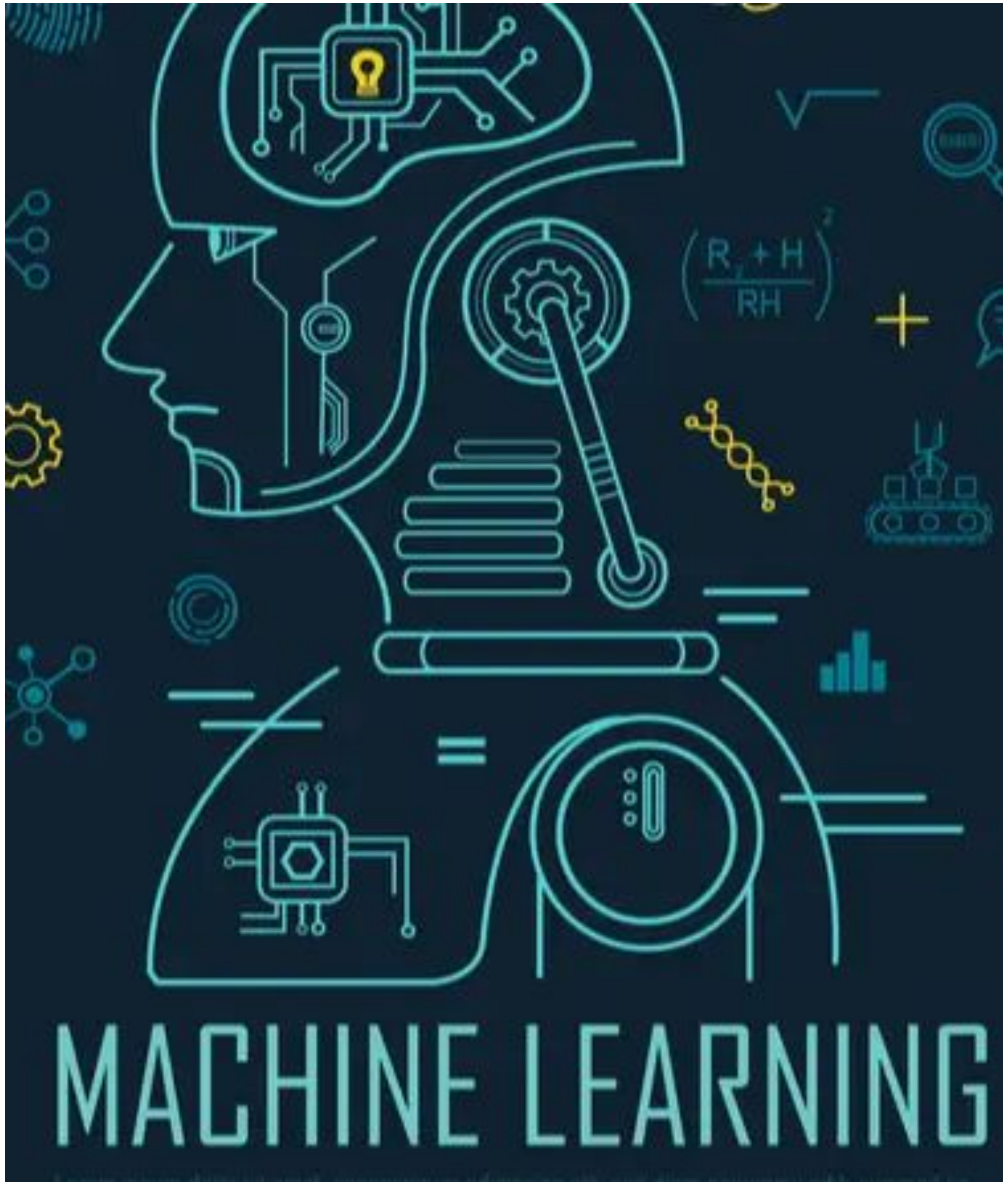
Lecture 3: Advanced Applications
- Multi-objective optimization
- Deep learning for defects
- Real-time process control

Types of machine learning

Supervised learning
- Learn from labeled examples: (input → output)
- Goal: predict output for new inputs
- AM Example: Process parameters → Part quality

Unsupervised learning
- Find patterns in unlabeled data
- Goal: discover structure
- AM Example: Group similar defects without predefined categories

Reinforcement Learning
- Learn through trial and error with rewards
- Goal: optimize sequential decisions
- AM Example: Adaptive path planning during printing



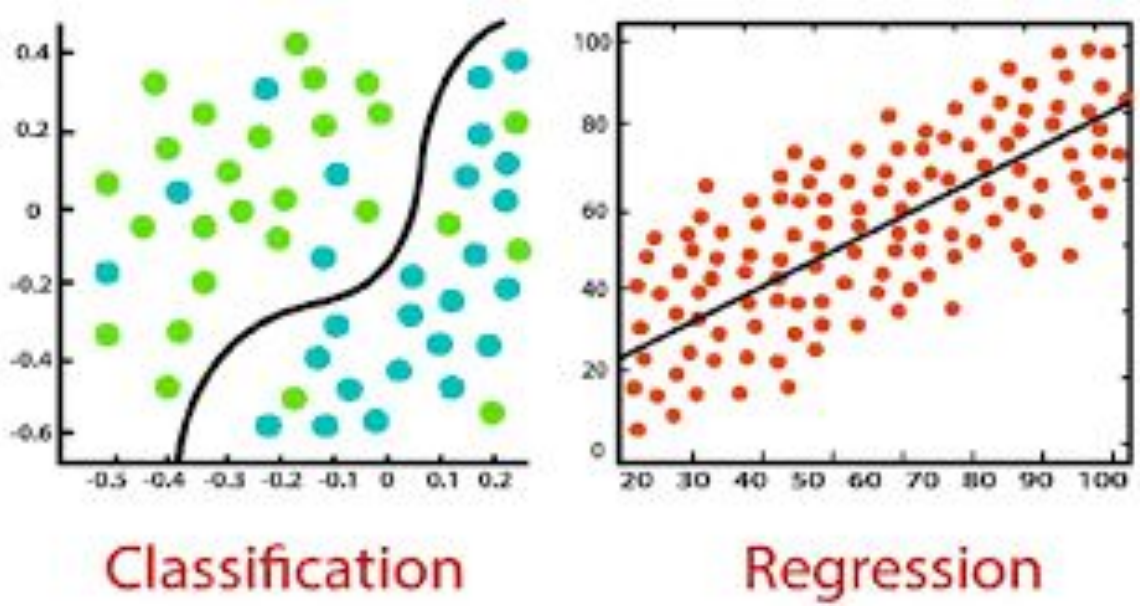
Supervised learning

Concept

- Given training data: $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$
- Learn function: $f(x) \approx y$

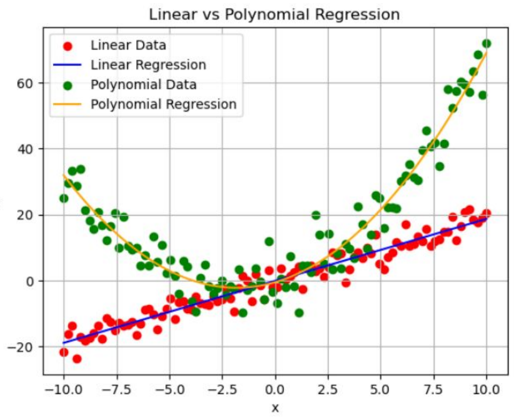
Flavours

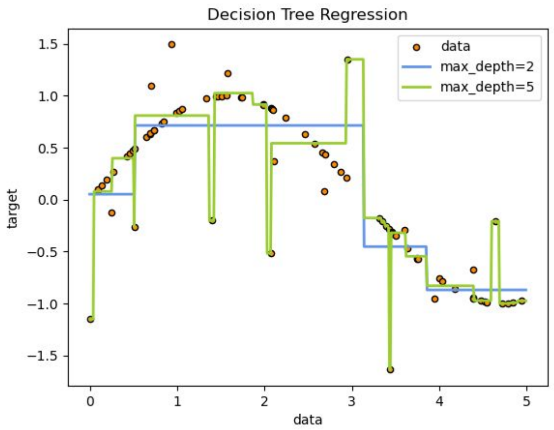
- Regression: predict numerical values
- Classification: predict category

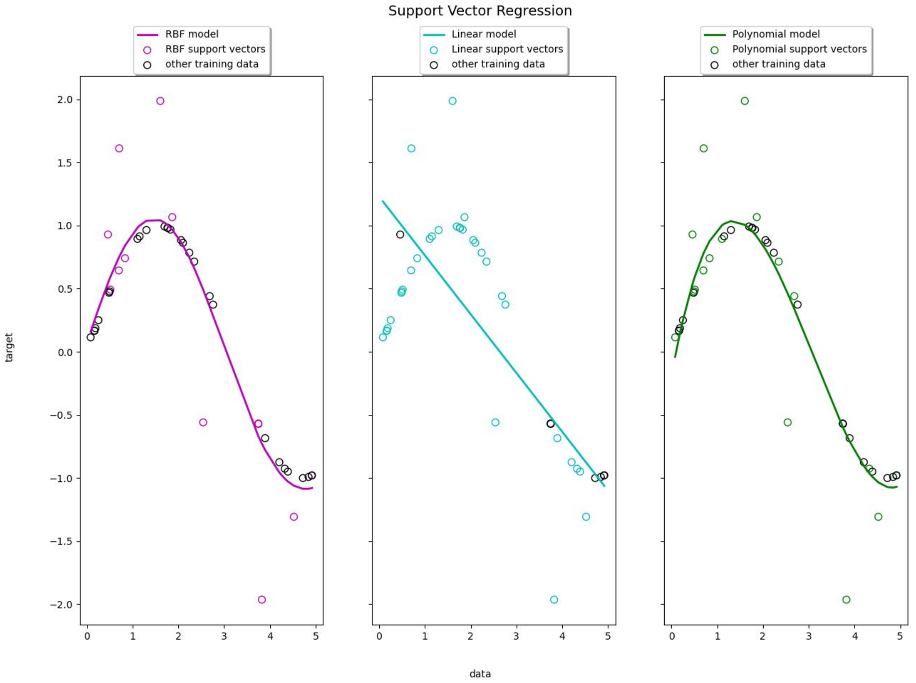


Key supervised learning algorithms

Algorithm	Characteristics
Linear Regression	• Simplest: fits a line/plane to data • Fast, interpretable • Limited to linear relationships
Decision Trees	• Series of if-then rules • Handles non-linearity • Can overfit easily
Random Forests	• Ensemble of decision trees • More robust, less overfitting • Good "default" choice for many AM problems
Support Vector Machines	• Finds optimal decision boundaries • Works well with limited data • Can handle high-dimensional spaces







Training vs Testing: never test on training data!

Common approach
- 70-80% Training set
- 10-15% Validation set
- 10-15% Test set

In manufacturing
- Consider temporal splits (train on old data, test on new)
- Account for different machines, operators, material batches
- Cross-validation for small datasets (train on k-1 folds, test on 1 fold)

Metrics for regression
- R^2 score (coefficient of determination)
- RMSE (Root Mean Square Error)
- MAE (Mean Absolute Error)

Metrics for Clasification
- Accuracy (careful with imbalanced data!)
- Precision, Recall, F1-score
- Confusion matrix
- ROC-AUC

❌ Wrong	✅ Correct
Normalize before splitting	Split first, then normalize (using only training statistics)
Feature selection on entire dataset	Feature selection only on training set
Using future information (time-series)	Respect temporal order

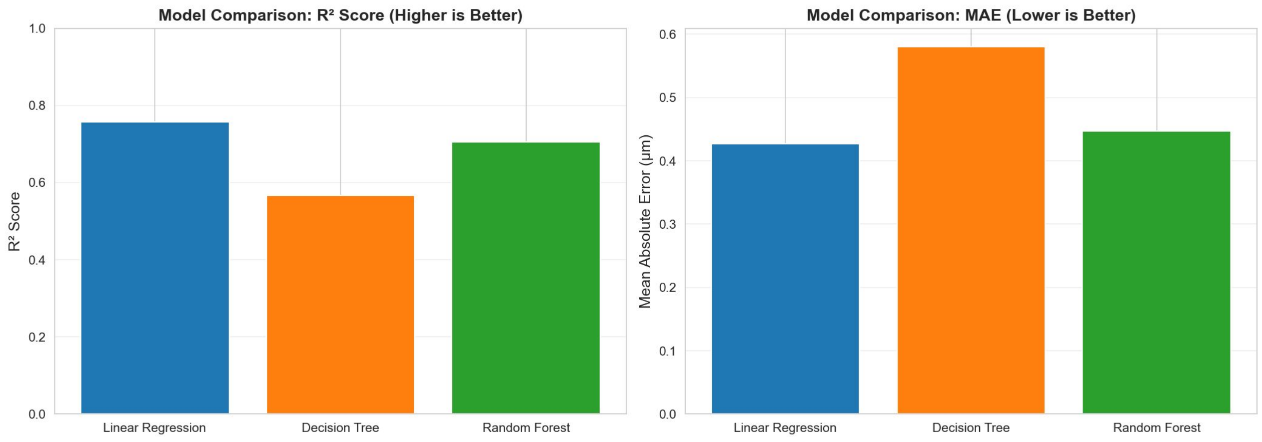
Demo 1: Predicting surface roughness

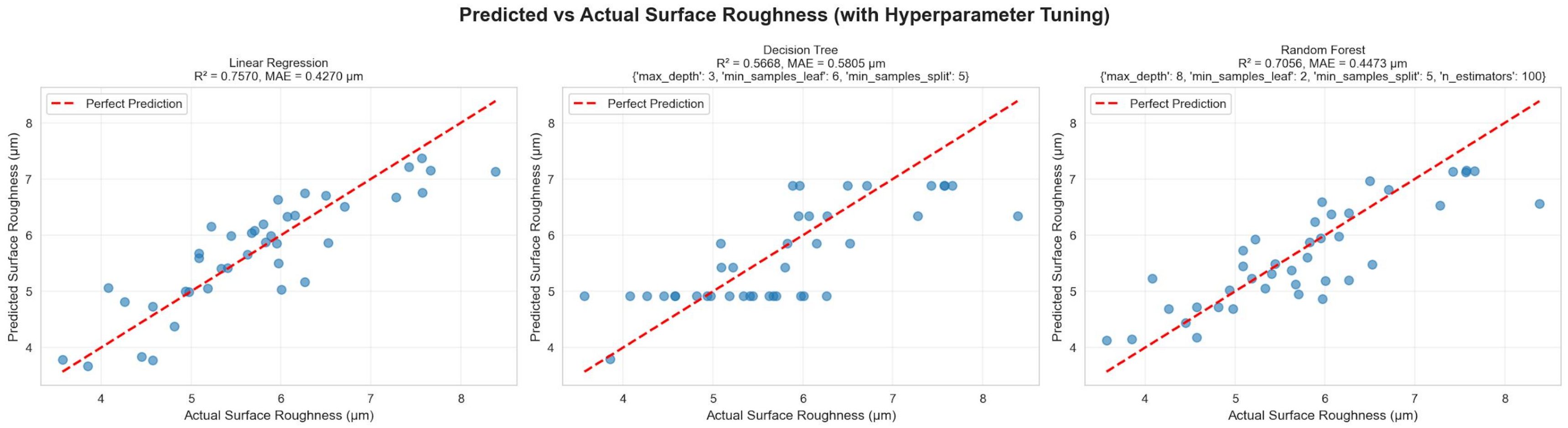
Problem

- Predict surface roughness from FDM process params

Input Features

- Layer height (mm)
- Print speed (mm/s)
- Nozzle temperature (°C)
- Bed temperature (°C)
- Infill density (%)



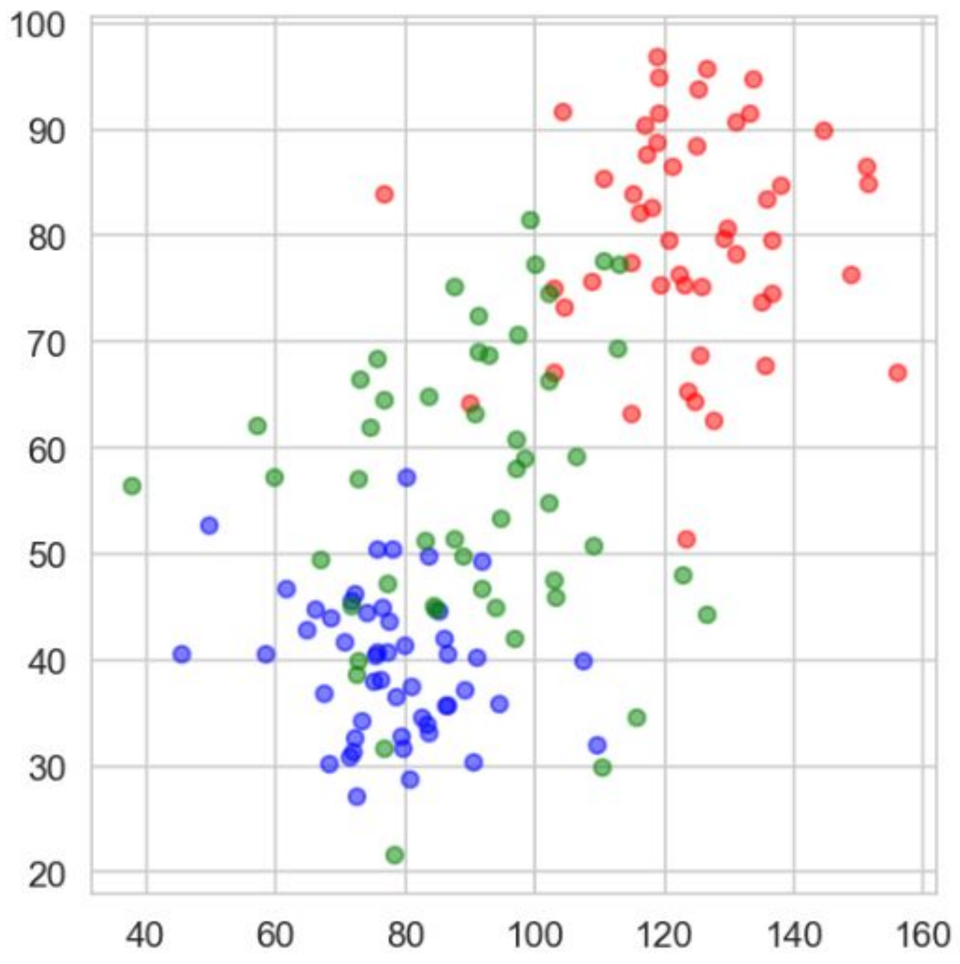


Unsupervised learning: Finding hidden patterns

Clustering
- Group similar observations together
- K-means, DBSCAN, Hierarchical clustering
- AM Example: Group build failures by similarity

Dimensionality Reduction
- Reduce number of features while preserving information
- PCA (Principal Component Analysis), t-SNE
- AM Example: Visualize high-dimensional sensor data

Anomaly Detection
- Identify unusual observations
- AM Example: Detect abnormal thermal signatures

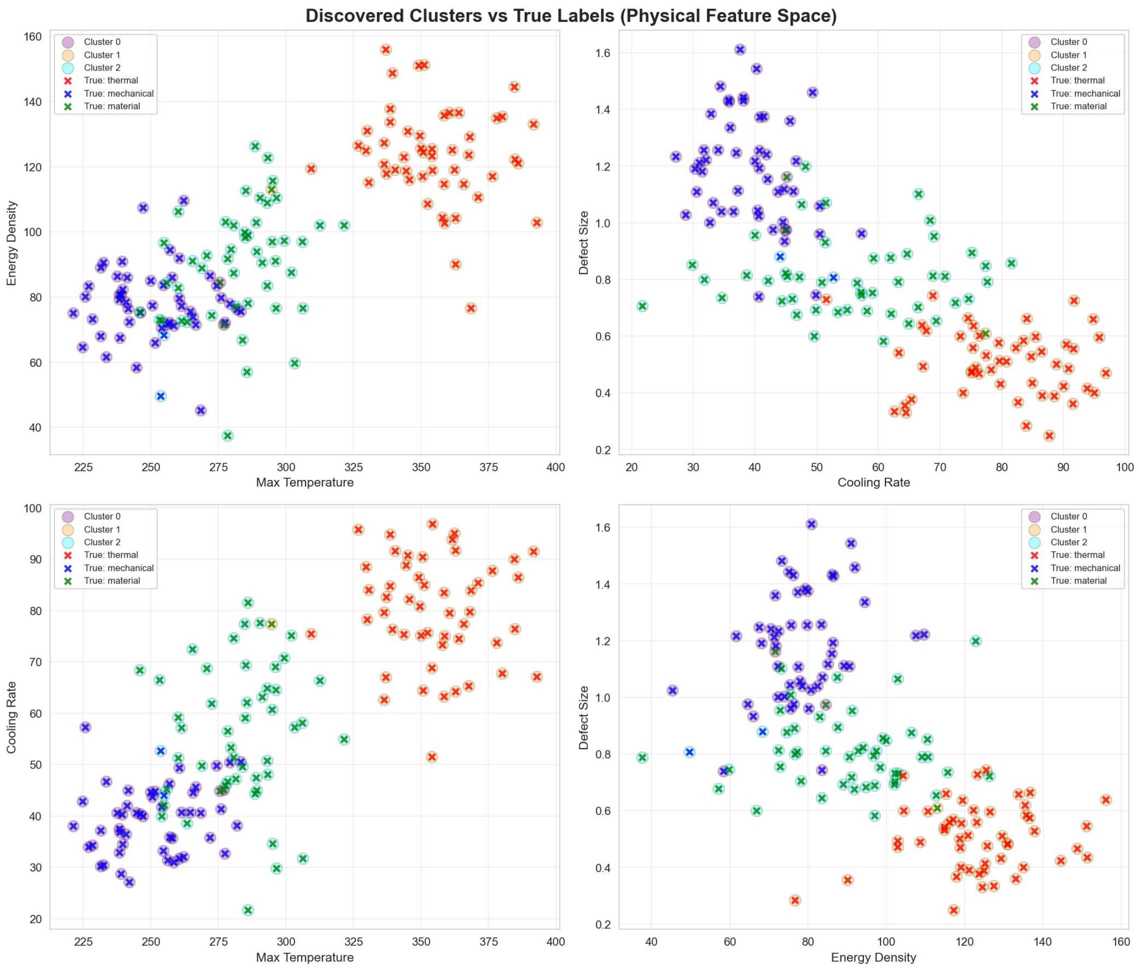


Demo 2: Deffect pattern discovery

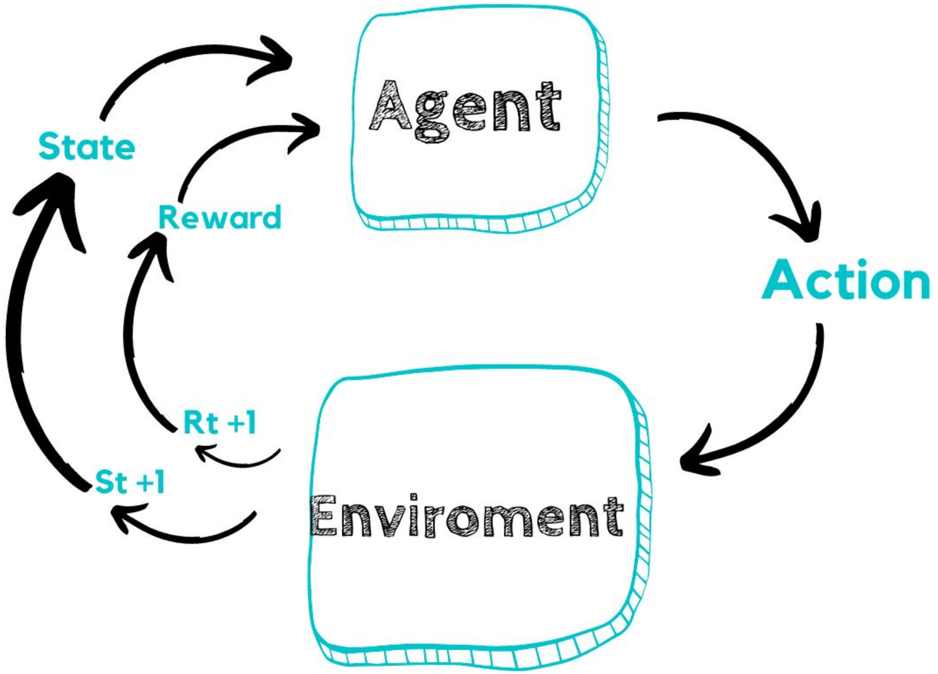
Problem
- Identify groups of similar defects without predefining categories

Input features
- Defect size, location, shape metrics
- Process conditions when defect occurred
- Material properties

Goal
- Discover natural groupings
- Maybe find: "thermal defects", "mechanical defects", "material defects"



Reinforcement learning: learning by doing



Key concepts
- Agent: The decision maker (e.g., print controller)
- Environment: The AM system
- State: Current condition (temperature, position, etc.)
- Action: Decision (adjust speed, change temperature)
- Reward: Feedback (quality improvement, time saved)

Goal
- Learn policy that maximizes cumulative reward

Example: Adaptive path planning in robocasting
- State: Current nozzle position, material flow rate, substrate condition
- Actions: Adjust travel speed, modify pressure, change path trajectory
- Reward: +1 for good layer adhesion, -1 for material discontinuity, -0.1 for time penalty

Note
- Very complex
- Expensive

Data sources in additive manufacturing

Pre-process data
- Material composition (ink formulation, powder chemistry & size)
- Rheological properties (viscosity, yield stress)
- Design parameters (geometry, infill, supports)
- Machine settings (temperature, speed, pressure)

In-process monitoring
- Thermal cameras (melt pool temperature)
- High-speed cameras (layer formation)
- Acoustic emission sensors (crack detection)
- Optical sensors (powder bed anomalies)

Post-process characterization
- CT scans (internal porosity)
- Surface profilometry (roughness)
- Mechanical testing (tensile strength, hardness)
- Microscopy (microstructure)

Process logs
- Machine parameters (as-commanded vs as-built)
- Environmental conditions
- Build statistics

Simulation data
- FEA (thermal/mechanical predictions)
- CFD (melt pool dynamics)
- Process simulations

Data quality challenge

Small sample sizes
- AM experiments are expensive
- Limited training data
- Risk of overfitting

Noisy measurements
- Sensor calibration drift
- Environmental interference
- Material variability

Imbalanced datasets
- Many good parts, few defects
- Difficult to learn failure modes

Missing data
- Sensor failures
- Incomplete measurements
- Historical data gaps

Multi-modal, data fusion
- Different sampling rates (thermal: 1000 Hz, mechanical: 1 Hz)
- Different scales and units
- Alignment challenges

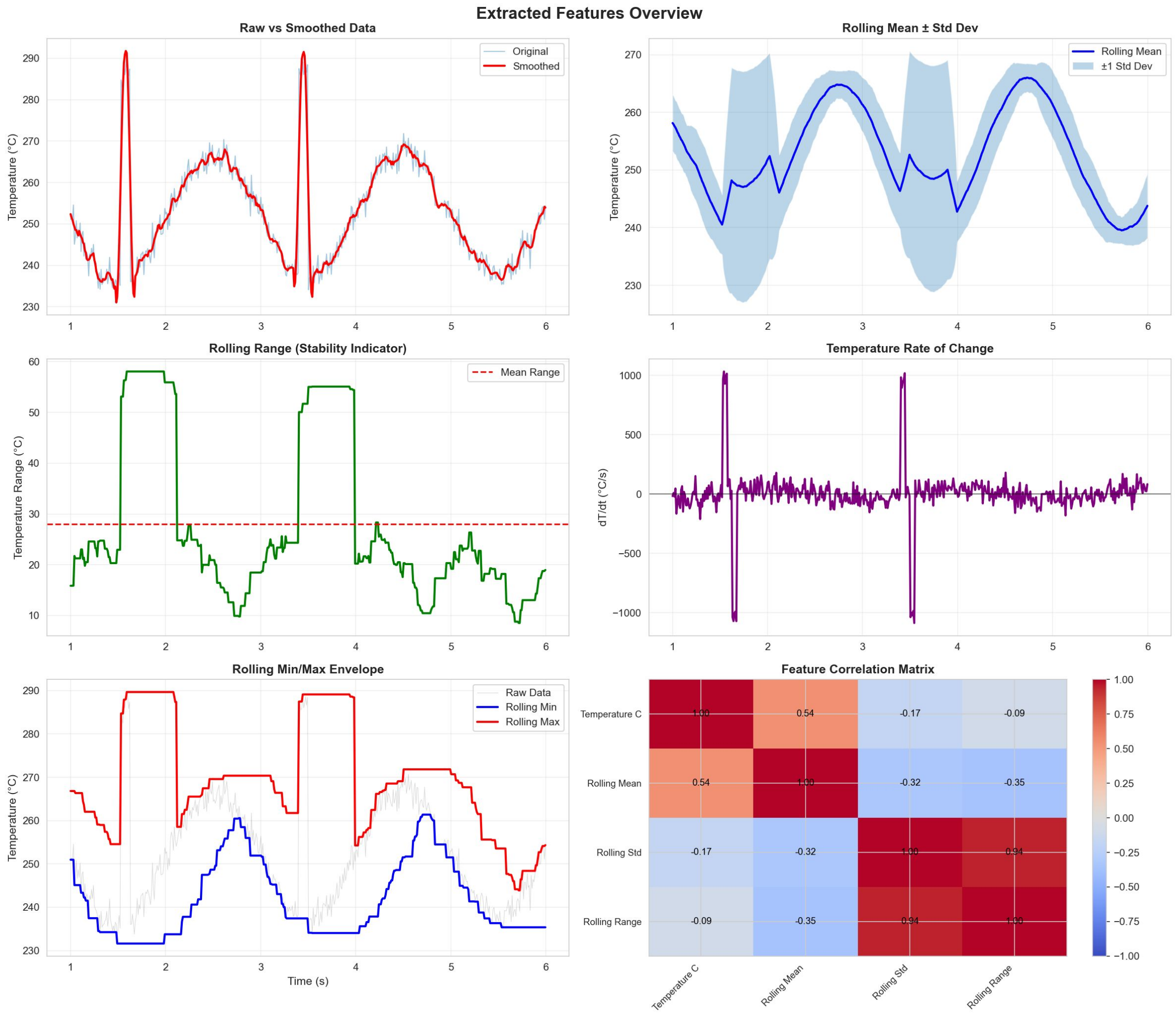
Demo 3: Data preprocessing

Data cleaning
- Handle missing values
- Remove outliers
- Fix inconsistencies

Normalization
- Scale features to comparable ranges
- StandardScaler: zero mean, unit variance
- MinMaxScaler: scale to [0, 1]

Feature extraction
- Time-series: statistical features
- Images: edge detection, texture
- Domain knowledge: energy density

Feature selection
- Remove redundant features
- Reduce dimensionality
- Improve model performance



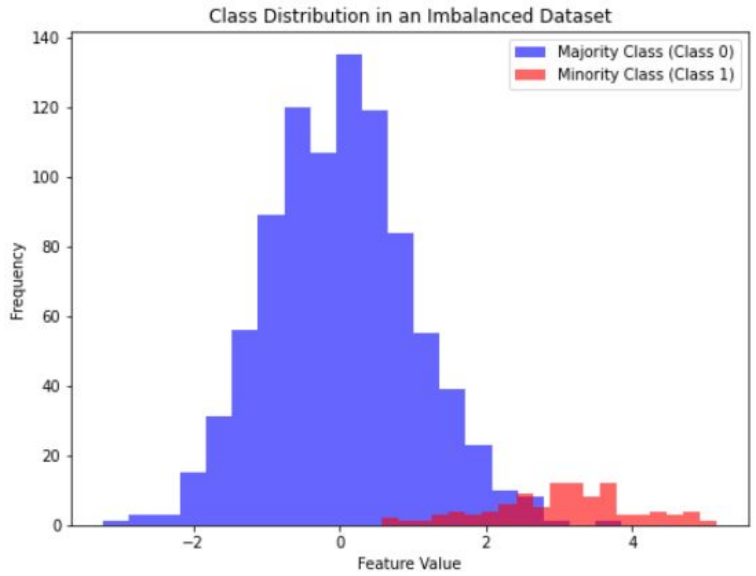
Handling imbalanced datasets in AM

The problem

- 95% good parts, 5% defective parts
- Model learns to predict "good" for everything
- Achieves 95% accuracy, but it is useless!

Solutions

- resampling: oversample minority class (SMOTE), undersample majority class
- class weights: penalize misclassification of rare classes heavily
- anomaly detection: treat defects as anomalies, one-class SVM, isolation forest
- synthetic data: use simulations to augment defect examples, GANs for data augmentation



Feature engineering for AM: domain knowledge is critical

Example 1: Thermal camera data
Raw: 1000×1000 pixels @ 50 Hz
Features:
- Max temperature
- Temperature gradient
- Cooling rate
- Melt pool area
- Temperature uniformity (std dev)

2. Process parameters
Features:
- Energy density: $E = P/(v \cdot h \cdot t)$
- Volumetric energy density: $VED = P/(v \cdot h \cdot d)$
- Cooling rate estimate
- Dimensionless numbers (Péclet, Reynolds)

Demo 4: The complete ML pipeline

1. Problem definition
- What are we predicting?
- What data do we need?
- What's good enough?

2. Data collection & EDA
- Gather data
- Exploratory analysis
- Understand distributions

3. Data preprocessing
- Clean data
- Normalize
- Engineer features

4. Model Selection
- Choose algorithm(s)
- Train on the training set
- Tune hyperparameters

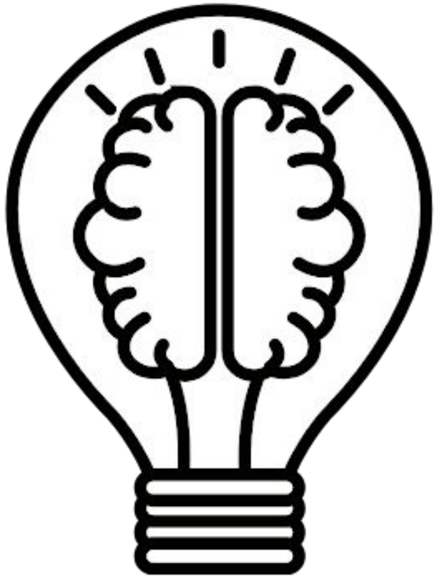
5. Validation & Testing
- Evaluate on unseen data
- Check for overfitting
- Compare models

6. Deployment
- Integrate with AM system
- Monitor performance
- Retrain as needed

Key takeaways

1. ML is about learning from data
- Supervised: learn input→output mappings
- Unsupervised: discover patterns
- Reinforcement: learn through trial and error

2. Data quality is critical
- Garbage in = garbage out
- AM data has unique challenges (small samples, noise, imbalance)
- Preprocessing is not optional!



3. ML pipeline should be systematic
- Problem definition → Data → Preprocessing → Model → Validation
- Avoid data leakage
- Always validate on unseen data

4. There is no “best” approach
- Choice depends on data size, interpretability needs, and resources
- Start simple, add complexity if needed

Common pitfalls

1. Overfitting with small datasets
- Model memorizes training data, fails on new data
- Solution: Use cross-validation, simpler models, regularization

2. Ignoring physics
- ML should complement, not replace, engineering knowledge
- Solution: Feature engineering benefits from domain expertise

3. Unrealistic expectations
- ML won't turn bad data into good predictions
- Solution: Needs sufficient, quality data

4. Not considering deployment
- Models trained offline may not work in real-time
- Solution: Computational constraints matter

Looking ahead to lecture 2: Process parameter opimization in AM

We'll dive deeper into:

- Regression techniques for continuous predictions
- Gaussian Process Regression for uncertainty quantification
- Bayesian Optimization for efficient parameter search
- Multi-fidelity approaches leveraging simulation + experiments