Creativity, Science and **Innovation**
Ideate, invent, innovate:
from ideas to prototypes

Time Series Preprocessing and Introduction to Machine Learning

November 20th, 2025

Susanna Bardini
susanna.bardini@polimi.it

Course Overview

13th November,12:30-14, Room 25.0.2
Understanding Physiological Data: Patterns, Correlations, and Explainability

20th November,12:30-14, Room 25.0.2
Time Series Preprocessing and Introduction to Machine Learning

24th November,12:30-14, Room 3.1.3
Machine Learning for Biomedical Tasks

1st December,12:30-14, Room 3.1.3
Introduction to Federated Learning

4th December,12:30-14, Room 25.0.2
Federated Learning: Practical Examples

Course Overview

13th November,12:30-14, Room 25.0.2
Understanding Physiological Data: Patterns, Correlations, and Explainability

20th November,12:30-14, Room 25.0.2
Time Series Preprocessing and Introduction to Machine Learning

24th November,12:30-14, Room 3.1.3 or https://politecnicomilano.webex.com/meet/alessandro.verosimile
Machine Learning for Biomedical Tasks

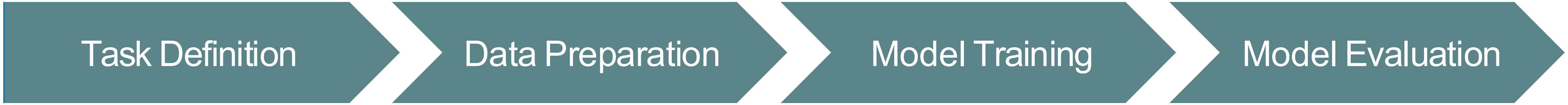
1st December,12:30-14, Room 3.1.3
Introduction to Federated Learning

4th December,12:30-14, Room 25.0.2
Federated Learning: Practical Examples

Course Material

https://github.com/BSusanna/CSI-TrackI-sem1-aa-2025-26.git

ML Model for Time Series/Signals Pipeline:



Task Definition

-Classification (binary, multi-label)

-Regression

Data Preparation

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

Model Training

-Decision Trees, SVM, Regression...

-RNN, GRU, LSTM

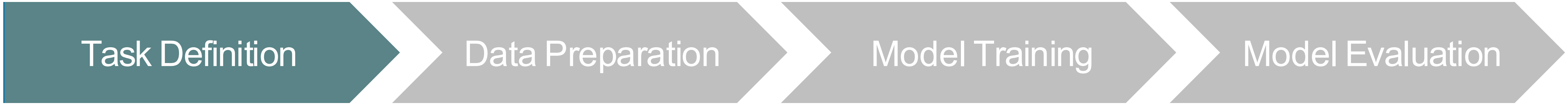
Model Evaluation

- Train / Test split

-Cross-validation

-LOOCV

ML Model for Time Series/Signals Pipeline:



Task Definition

- -Classification (binary, multi-label)
- -Regression

Data Preparation

- -Handling abnormalities
- -Data/Signal Processing
- -Feature Extraction
- -Dimensionality Reduction

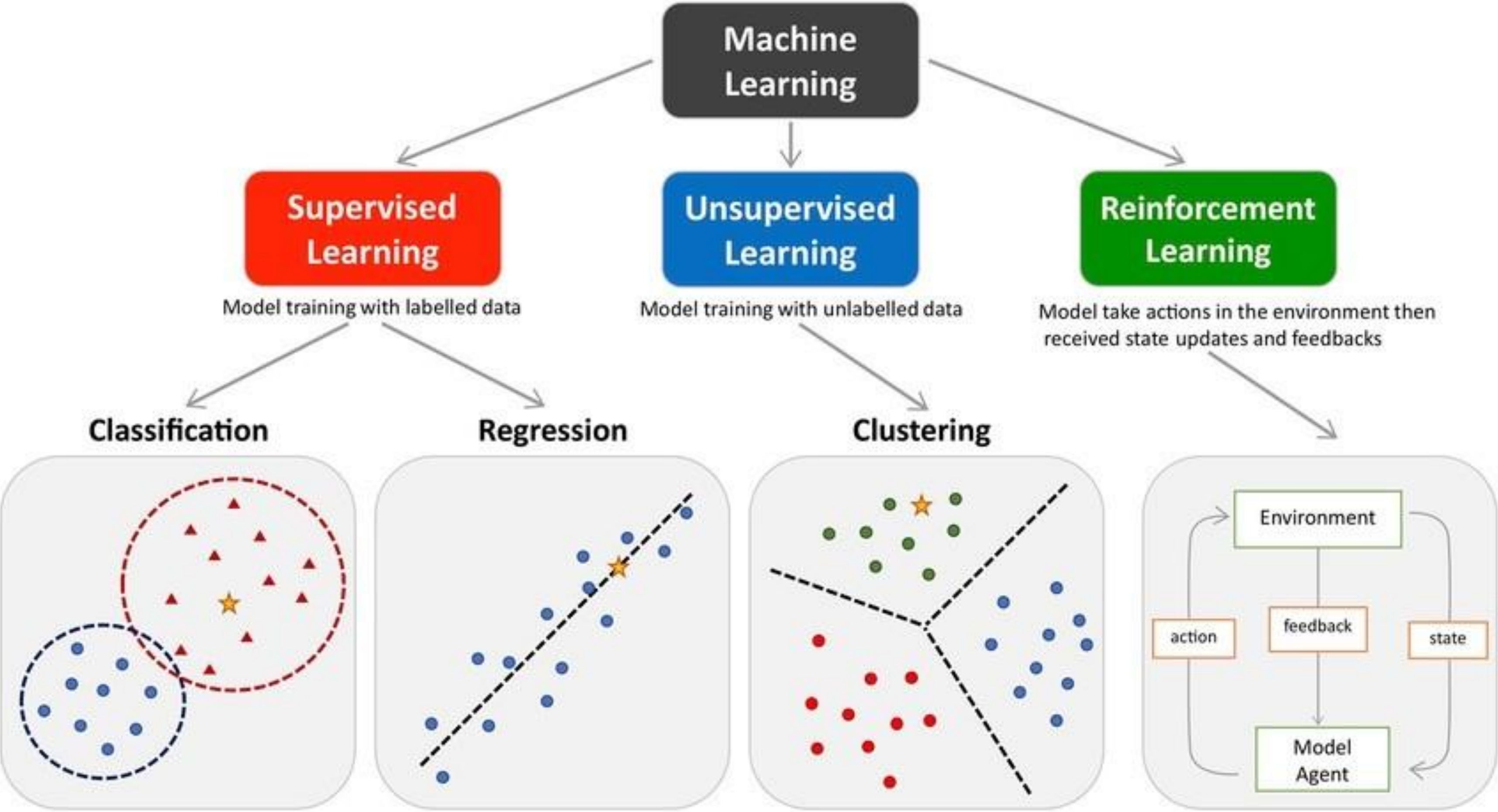
Model Training

- -Decision Trees, SVM, Regression...
- -RNN, GRU, LSTM

Model Evaluation

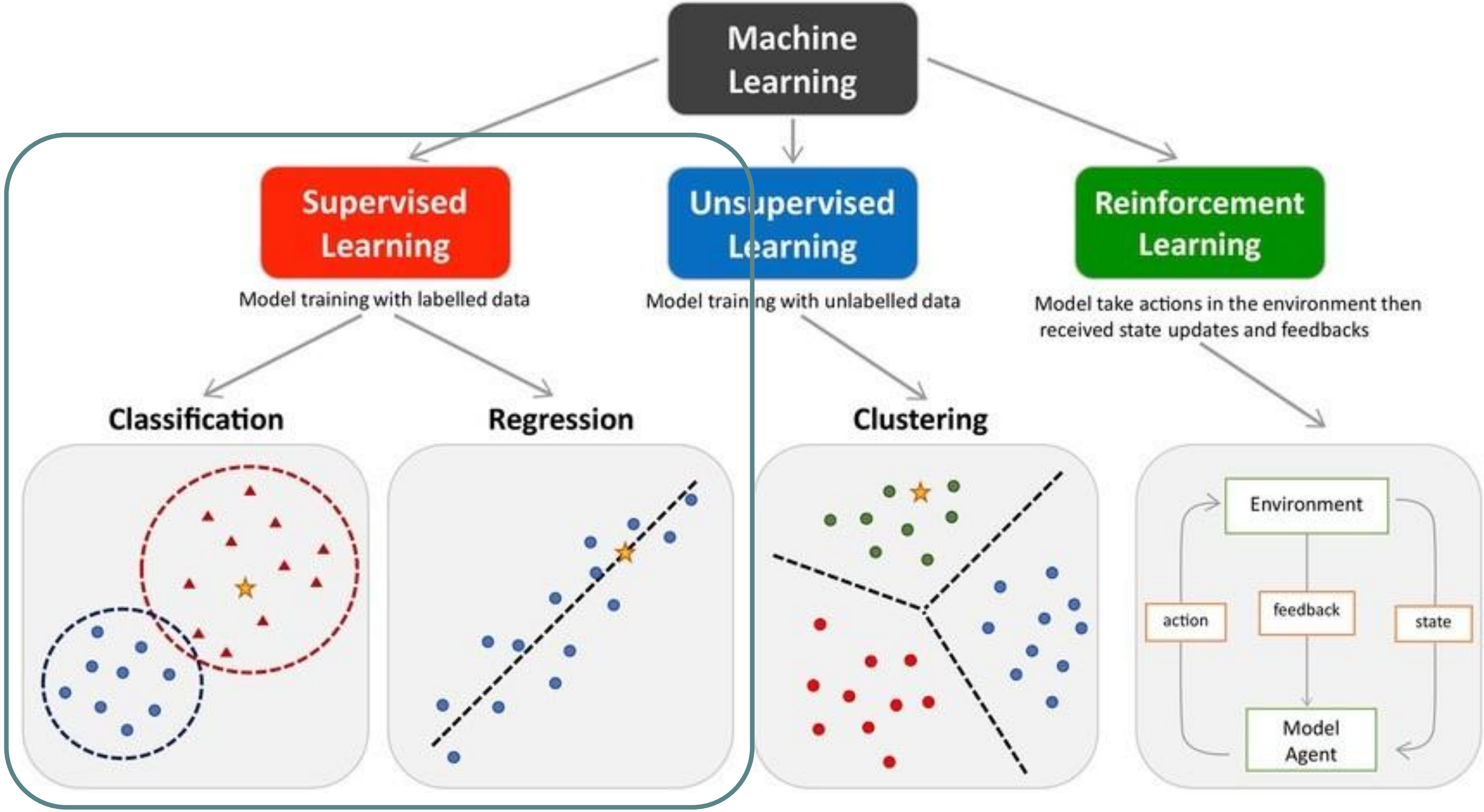
- - Train / Test split
- -Cross-validation
- -LOOCV

Machine Learning categories



Task Definition → Data Preperation → Model Training → Model Evaluation

Machine Learning categories



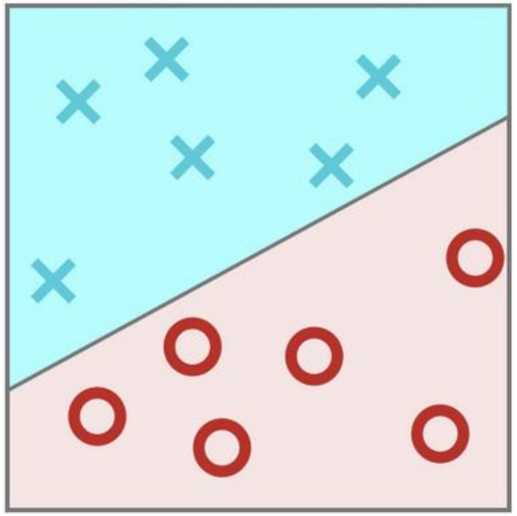
Task Definition → Data Preperation → Model Training → Model Evaluation

Classification vs Regression

Classification

Definition: Classification is a supervised machine learning task where the model learns to predict discrete labels or categories based on input data.

Example: Suppose you want to classify emails as either "spam" or "not spam." You train a classification model using labeled email. The model learns patterns (like specific keywords or sender details) and, once trained, can classify new emails as spam or not.

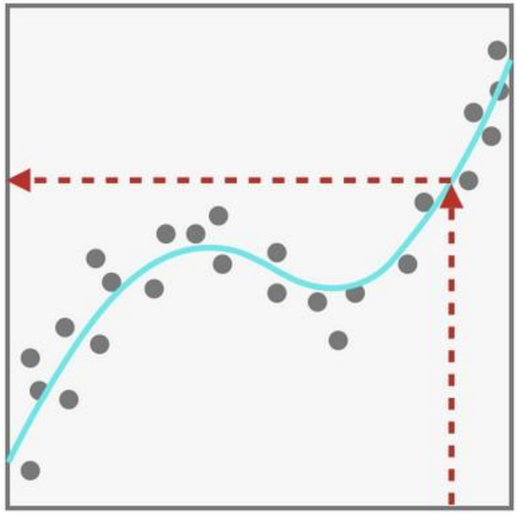


Classification vs Regression

Regression

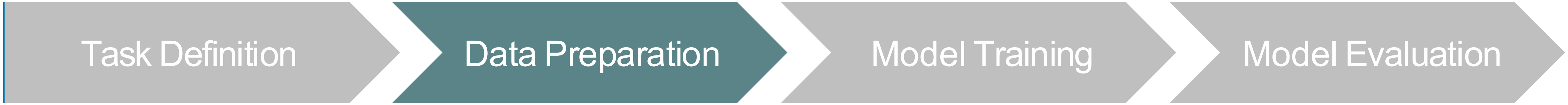
Definition: Regression is a supervised learning task where the model learns to predict a continuous value based on input data. Unlike classification, which outputs discrete labels, regression **outputs numerical values.**

Example: Suppose you want to predict the price of a house based on features like the number of bedrooms, square footage, and location. You train a regression model with data, then it learns relationships between features and prices, so that it can predict the price of new houses based on features.



Task Definition → Data Preparation → Model Training → Model Evaluation

ML Model for Time Series/Signals Pipeline:



Task Definition

Data Preparation

Model Training

Model Evaluation

-Classification (binary, multi-label)

-Regression

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

-Decision Trees, SVM, Regression...

-RNN, GRU, LSTM

- Train / Test split

-Cross-validation

-LOOCV

Handling Abnormalities

Time (s)	X-Axis Acceleration (m/s²)	Y-Axis Acceleration (m/s²)	Z-Axis Acceleration (m/s²)
0.0	0.1	9.8	-0.3
0.2	NaN	9.91	-0.2
0.4	0.1	NaN	0.0
0.6	0.4	9.8	0.1
0.8	0.3	NaN	0.1
1.0	0.2	9.75	0.6
1.2	0.3	9.7	NaN

Task Definition → **Data Preparation** → Model Training → Model Evaluation

Handling Abnormalities

Time (s)	X-Axis Acceleration (m/s²)	Y-Axis Acceleration (m/s²)	Z-Axis Acceleration (m/s²)
0.0	0.1	9.8	-0.3
0.2	NaN	9.91	-0.2
0.4	0.1	NaN	0.0
0.6	0.4	9.8	0.1
0.8	0.3	NaN	0.1
1.0	0.2	9.75	0.6
1.2	0.3	9.7	NaN

Handling Abnormalities

1. Forward/Backward Fill: Use the last known value (forward fill) or next known value (backward fill) to replace missing data, maintaining continuity.

Y-Axis Acceleration (m/s²)
9.8
9.91
NaN
9.8
NaN
9.75
9.7

Handling Abnormalities

1. Forward/Backward Fill: Use the last known value (forward fill) or next known value (backward fill) to replace missing data, maintaining continuity.

Forward Fill

Y-Axis Acceleration (m/s²)

9.8

9.91

↓

9.91

9.8

↓

9.8

9.75

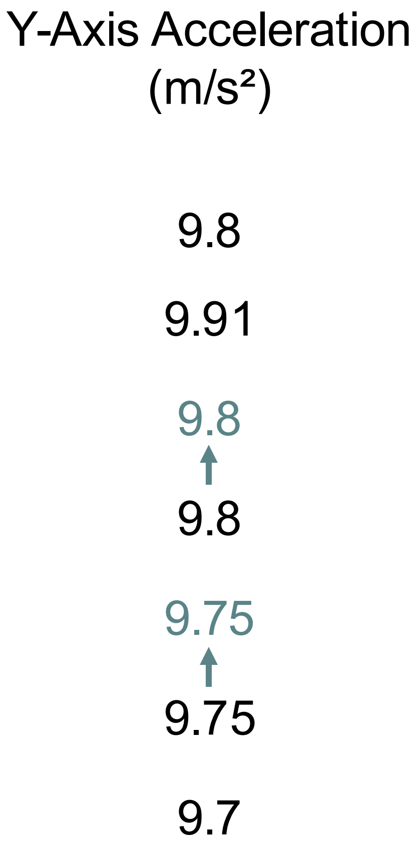
9.7

Task Definition → Data Preparation → Model Training → Model Evaluation

Handling Abnormalities

1. Forward/Backward Fill: Use the last known value (forward fill) or next known value (backward fill) to replace missing data, maintaining continuity.

Backward Fill



Y-Axis Acceleration (m/s²)

9.8

9.91

9.8

9.8

9.75

9.75

9.7

Task Definition → Data Preparation → Model Training → Model Evaluation

Handling Abnormalities

1. Forward/Backward Fill: Use the last known value (forward fill) or next known value (backward fill) to replace missing data, maintaining continuity.

2. Linear Interpolation: Estimate missing values by interpolating between adjacent data points, providing a smooth transition in the time series.

Y-Axis Acceleration (m/s²)
9.8
9.91
9.855
9.8
9.775
9.75
9.7

Task Definition → **Data Preparation** → Model Training → Model Evaluation

Handling Abnormalities

3. Seasonal Interpolation: For seasonal data, use values from the same time in previous cycles (e.g., previous years) to estimate missing values.

Y-Axis Acceleration (m/s^2)
9.8
9.91
NaN
9.8
NaN
9.75
9.7

Handling Abnormalities

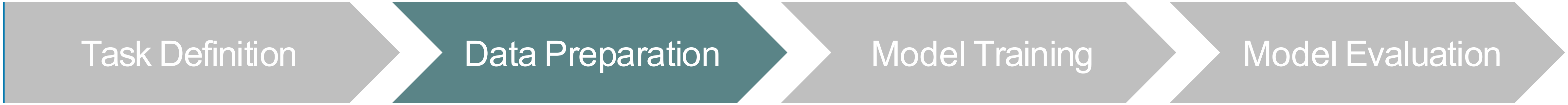
3. Seasonal Interpolation: For seasonal data, use values from the same time in previous cycles (e.g., previous years) to estimate missing values.

4. Advanced Methods: Techniques like Kalman Filtering, ARIMA Modeling, and Multiple Imputation offer more sophisticated estimates:

ARIMA (AutoRegressive Integrated Moving Average) models can predict missing values based on trends, seasonality, and autocorrelation patterns within the time series. This approach is particularly effective for stationary series with recurring patterns.

Y-Axis Acceleration (m/s²)
9.8
9.91
NaN
9.8
NaN
9.75
9.7

ML Model for Time Series/Signals Pipeline:



Task Definition

Data Preparation

Model Training

Model Evaluation

-Classification (binary, multi-label)

-Regression

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

-Decision Trees, SVM, Regression...

-RNN, GRU, LSTM

- Train / Test split

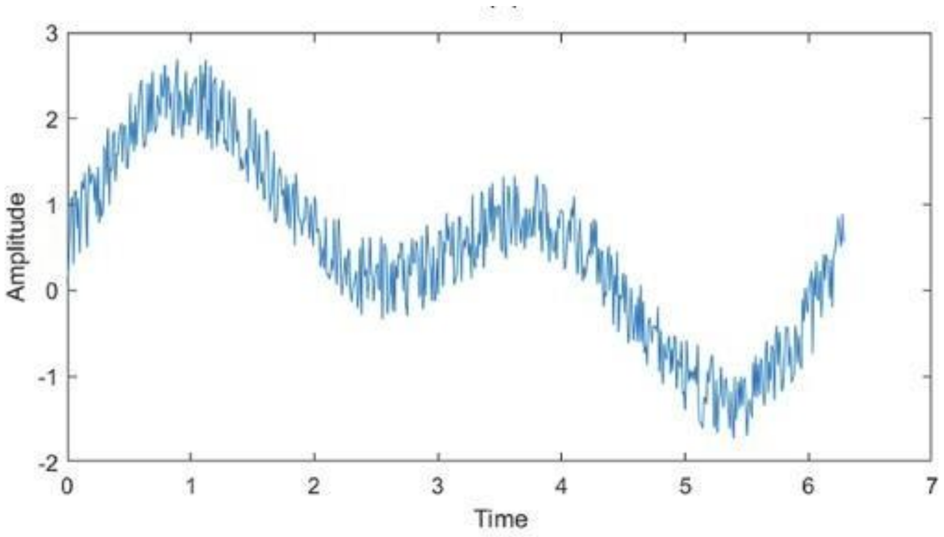
-Cross-validation

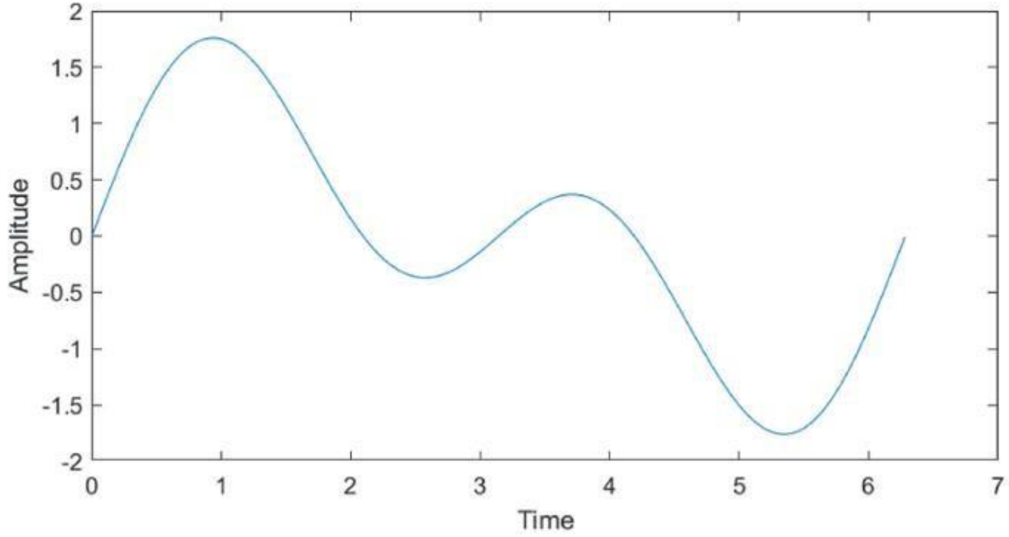
-LOOCV

TS/Signal Processing

1. Noise Removal

Noise represents a variation of the signal that negatively impacts ML models performance. Noise is in general associated with high frequencies.



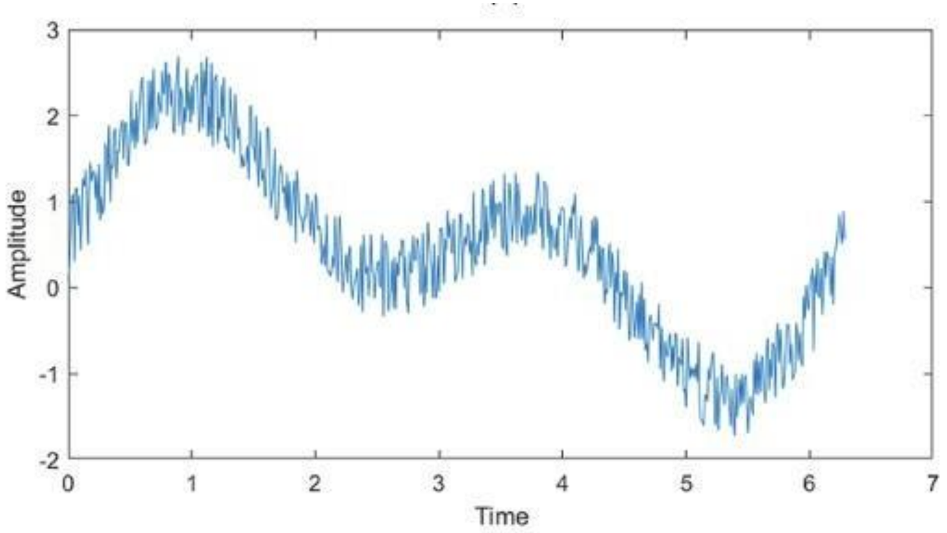


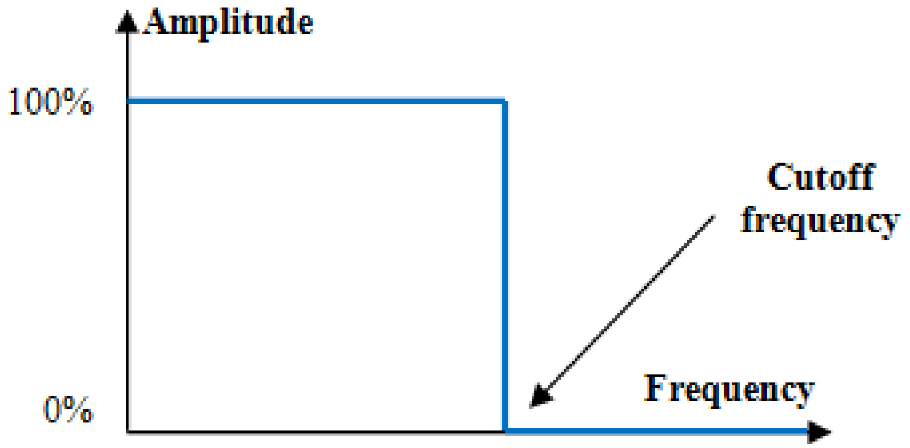
Purpose: To remove unwanted variations or random fluctuations that could distort our analysis.

Task Definition → **Data Preparation** → Model Training → Model Evaluation

TS/Signal Processing

1. Noise Removal



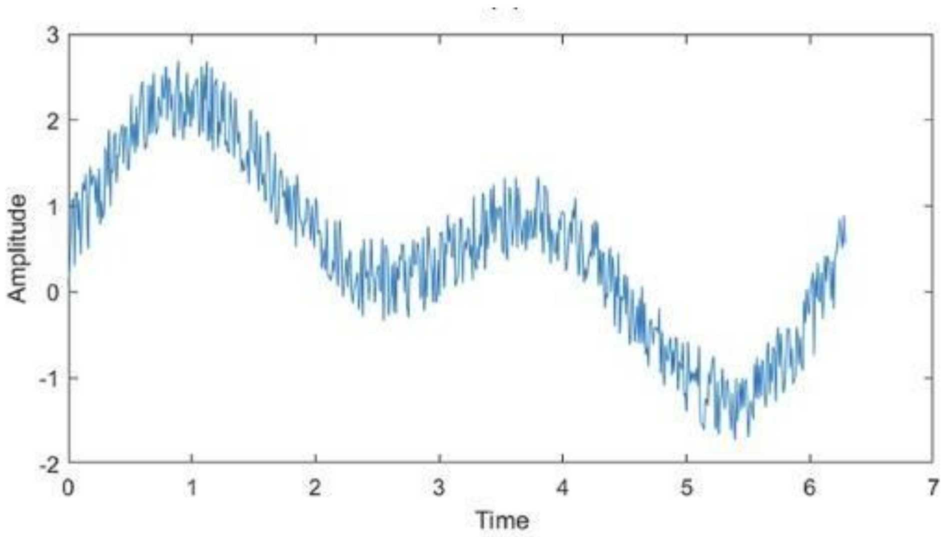


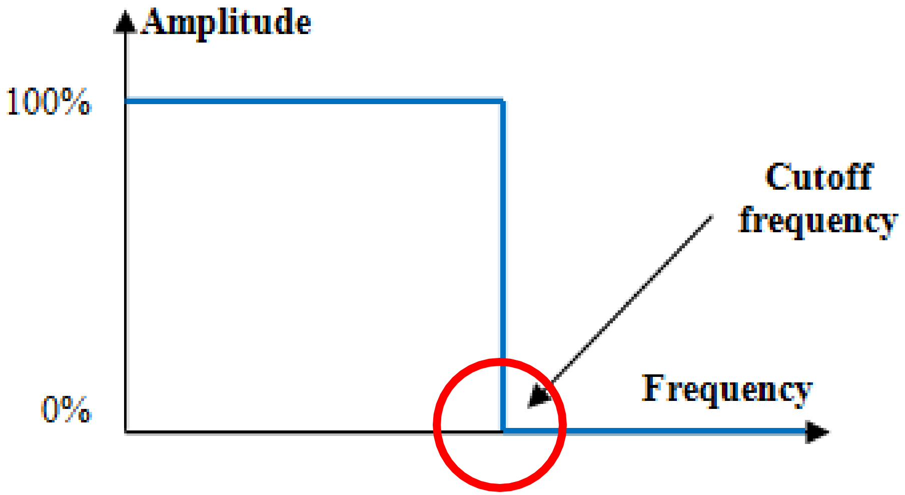
A low-pass filter is a tool that allows low-frequency signals (like slow changes or steady trends) to pass through while blocking high-frequency signals (like sudden spikes or noise). In simple terms, **it smooths out a signal by removing its rapid fluctuations**

Task Definition | **Data Preparation** | Model Training | Model Evaluation

TS/Signal Processing

1. Noise Removal





The cutoff frequency is the point where a filter starts to block or reduce certain parts of a signal. **For a low-pass filter, the cutoff frequency is the highest frequency that the filter will allow through**. Everything above this frequency gets reduced or blocked

TS/Signal Processing

2. Standardization/Normalization of the Signal/TS

- Standardizing: Adjusts data to have a mean of 0 and a standard deviation of 1, making it follow a standard scale without restricting its range.

- Normalizing: Scales data to fit within a fixed range, usually between 0 and 1, making values easier to compare directly.

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma}$$

$$x_{\text{normalized}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

Task Definition

Data Preparation

Model Training

Model Evaluation

TS/Signal Processing

2. Standardization/Normalization of the Signal/TS

	Standardization	Normalization
Definition	Adjusts data to have a mean of 0 and a standard deviation of 1	Scales data to fit within a fixed range, typically 0 to 1
Range of Values	Unbounded	Fixed range (e.g., 0 to 1)
When to use	When data has a normal distribution or wide range of values	More sensitive (outliers impact min and max)
Applications	Useful for algorithms assuming normal distribution (e.g., SVM, linear regression)	Common in neural networks, image processing, or distance-based models (e.g., KNN)

Task Definition

Data Preparation

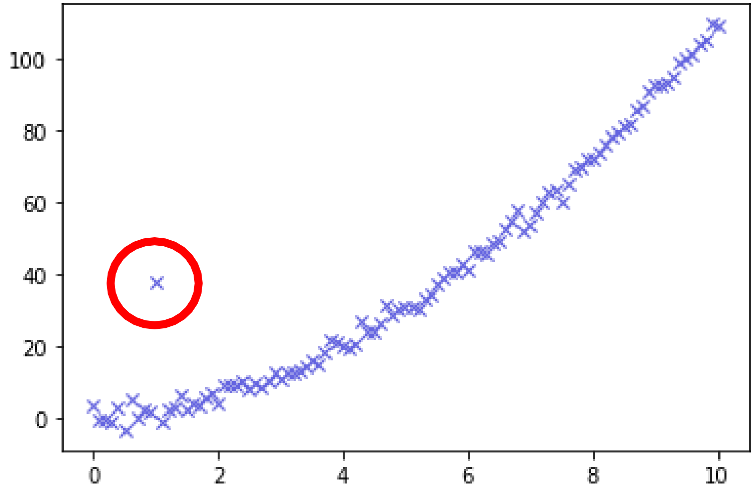
Model Training

Model Evaluation

TS/Signal Processing

3. Outlier Detection and Removal

An outlier is a data point that significantly deviates from the other observations in a dataset. Outliers often occur due to measurement errors, data entry mistakes, or rare events that don't represent the typical pattern



Purpose: Removing outliers can improve model performance, especially in cases where the outlier doesn't represent the general behavior of the data.

Task Definition → Data Preparation → Model Training → Model Evaluation

TS/Signal Processing

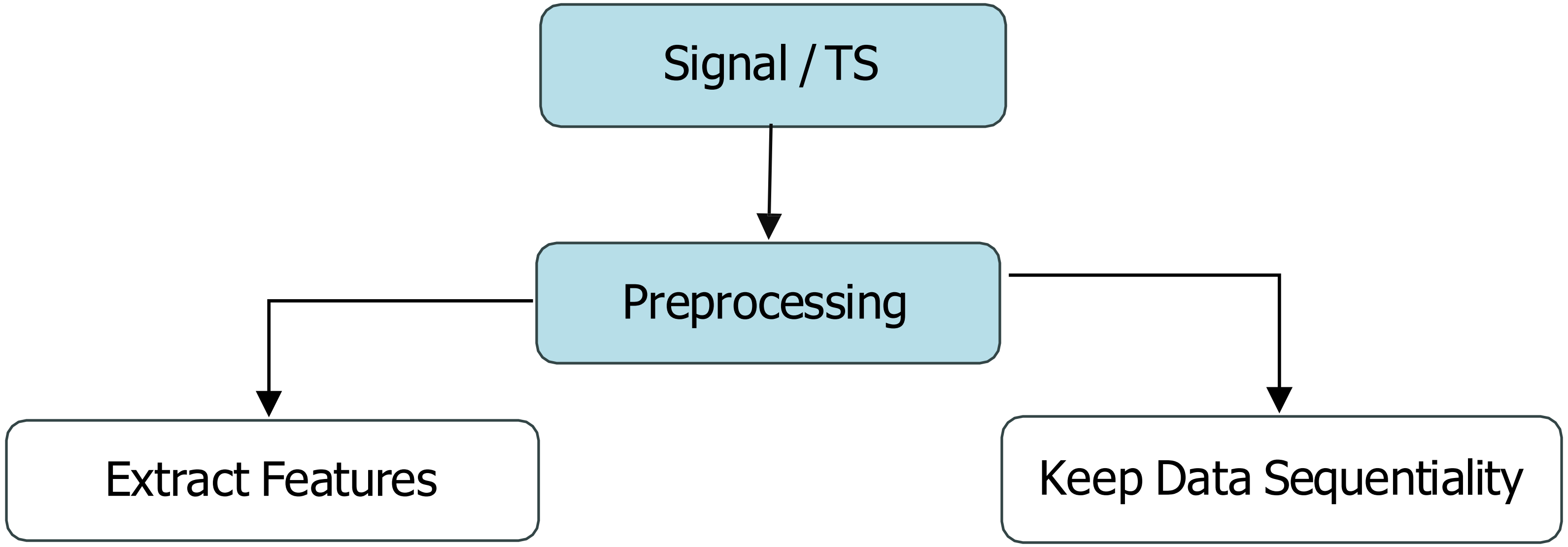
3. Outlier Detection and Removal

- Standard Deviation Method: Identify points that are more than a set number of standard deviations (e.g., 3) from the mean. Remove these extreme values to reduce their impact on analysis.

- Interquartile Range (IQR) Method: Calculate Q1 and Q3, then find the IQR (Q3 - Q1). Outliers are points below $Q1 - 1.5 * IQR$ and above $Q3 + 1.5 * IQR$ and they can therefore be removed.

- Z-Score Method: Compute the z-score for each point (how many standard deviations it is from the mean). Points with a z-score above 3 or below -3 are typically considered outliers and can be removed.

Example

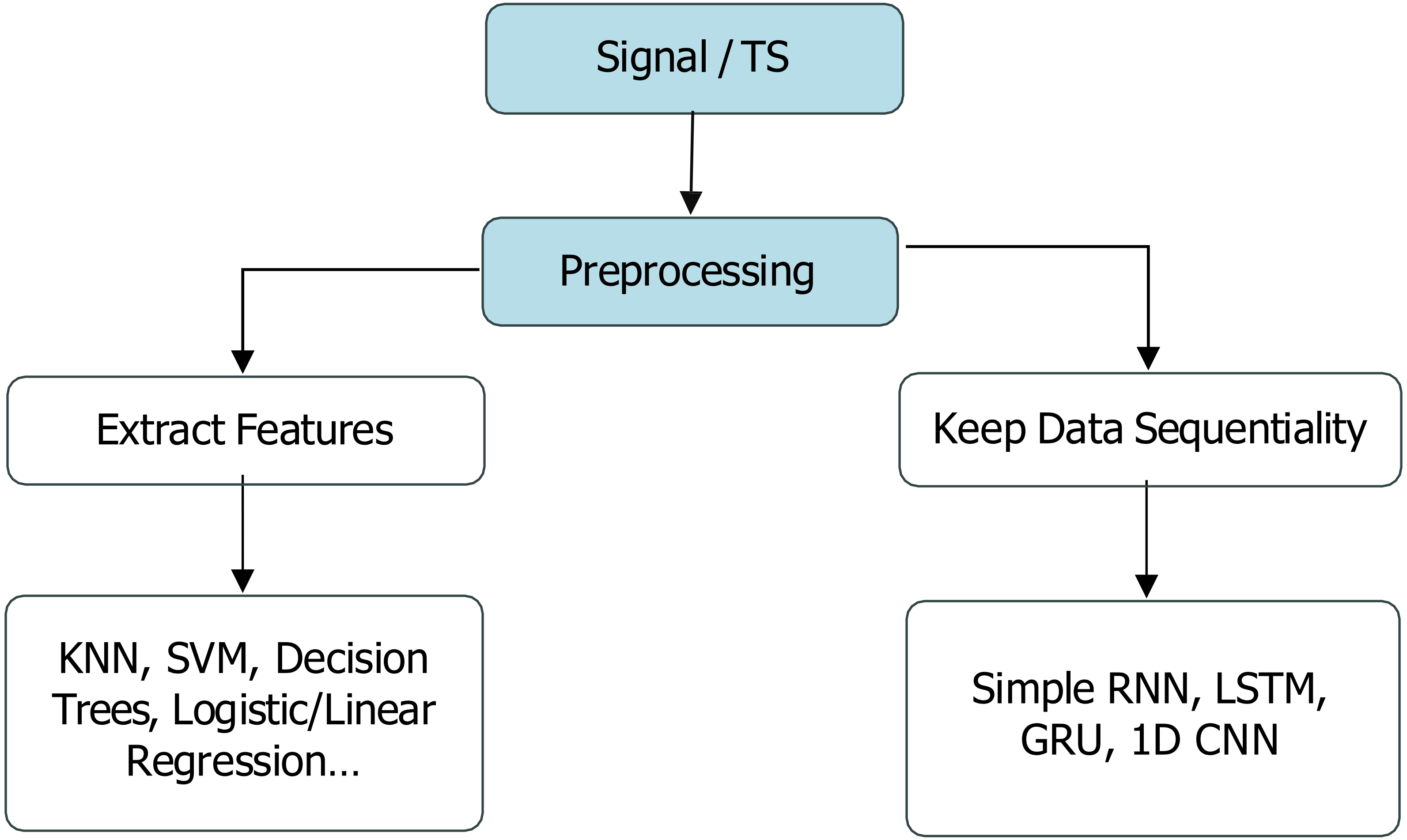


Signal / TS

Preprocessing

Extract Features

Keep Data Sequentiality



Signal / TS

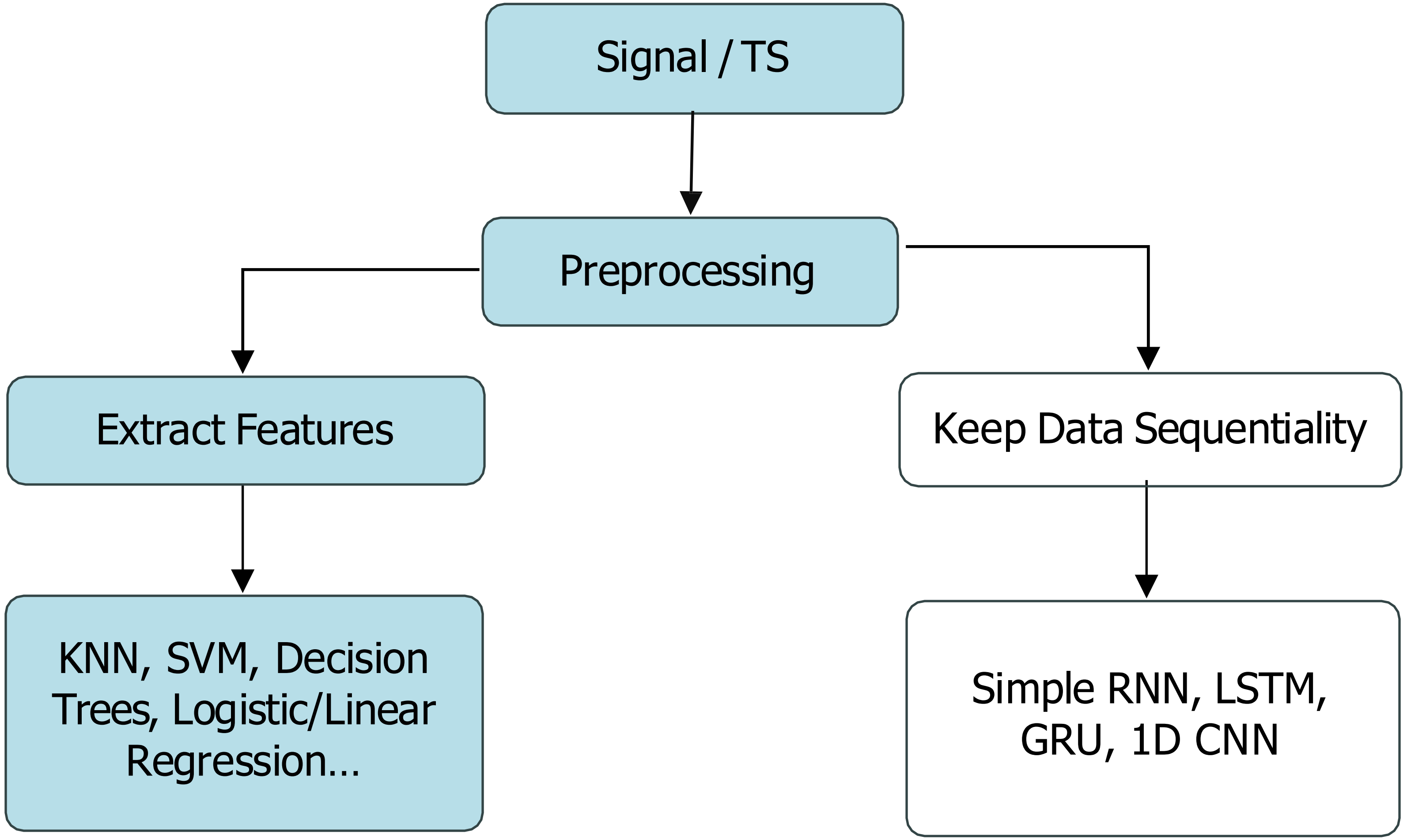
Preprocessing

Extract Features

Keep Data Sequentiality

KNN, SVM, Decision Trees, Logistic/Linear Regression…

Simple RNN, LSTM, GRU, 1D CNN



Signal / TS

Preprocessing

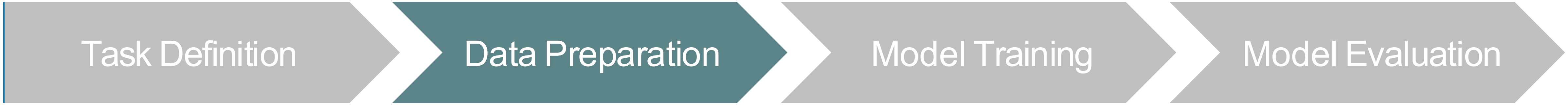
Extract Features

Keep Data Sequentiality

KNN, SVM, Decision Trees, Logistic/Linear Regression…

Simple RNN, LSTM, GRU, 1D CNN

ML Model for Time Series/Signals Pipeline:



Task Definition

-Classification (binary, multi-label)

-Regression

Data Preparation

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

Model Training

-Decision Trees, SVM, Regression...

-RNN, GRU, LSTM

Model Evaluation

- Train / Test split

-Cross-validation

-LOOCV

Feature Extraction

A few examples:

1. Statistical Features

- Mean: Average value of the signal.
- Standard Deviation: Measures the spread of values around the mean.
- Variance: Degree of variation in the signal.
- Skewness: Indicates the asymmetry of the signal distribution.
- Kurtosis: Measures the "tailedness" or sharpness of the peak of the distribution.

2. Temporal Features

- Peak-to-Peak Amplitude: Difference between the maximum and minimum values in the signal.
- Autocorrelation: Correlation of the signal with itself at different time lags.
- Zero-Crossing Rate: Number of times the signal crosses the zero line, indicating frequency content.
- Number of Peaks: Count of significant peaks in the signal, often related to activity frequency.

Feature Extraction

A few examples:

3. Time-Frequency Features

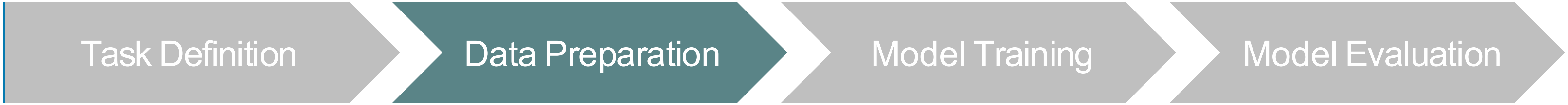
- Wavelet Coefficients: Extracted to capture both time and frequency information.
- Short-Time Fourier Transform (STFT): Breaks down the signal into segments, providing frequency information over time
- Energy Entropy: Measures the distribution of energy across different time-frequency components

4. Shape-Based Features

- Slope: Rate of change in the signal over time.
- Linearity: How closely the signal follows a straight line.
- Area Under Curve (AUC): Total area between the signal and the zero line, related to cumulative activity.

And other time-domain, frequency-domain, and domain-specific features.

ML Model for Time Series/Signals Pipeline:



Task Definition

Data Preparation

Model Training

Model Evaluation

-Classification (binary, multi-label)

-Regression

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

-Decision Trees, SVM, Regression...

-RNN, GRU, LSTM

- Train / Test split

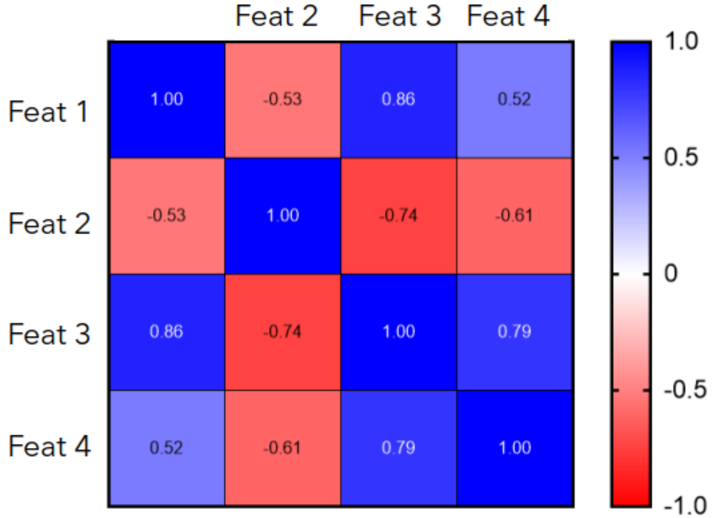
-Cross-validation

-LOOCV

Dimensionality Reduction: Feature Selection

Feature selection is essential for improving model accuracy and reducing complexity by selecting only the most relevant features. Here are the main methods:

- Filter methods: Significance-based feature selection

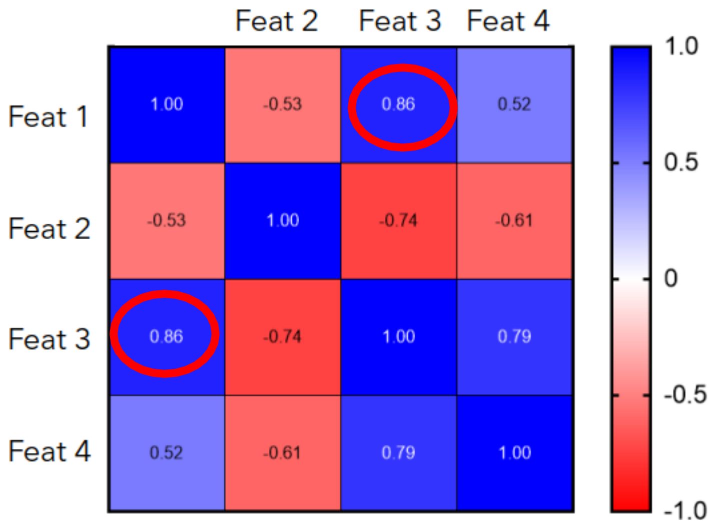


		Feat 2	Feat 3	Feat 4
Feat 1	1.00	-0.53	0.86	0.52
Feat 2	-0.53	1.00	-0.74	-0.61
Feat 3	0.86	-0.74	1.00	0.79
Feat 4	0.52	-0.61	0.79	1.00

Dimensionality Reduction: Feature Selection

Feature selection is essential for improving model accuracy and reducing complexity by selecting only the most relevant features. Here are the main methods:

- Filter methods: Significance-based feature selection



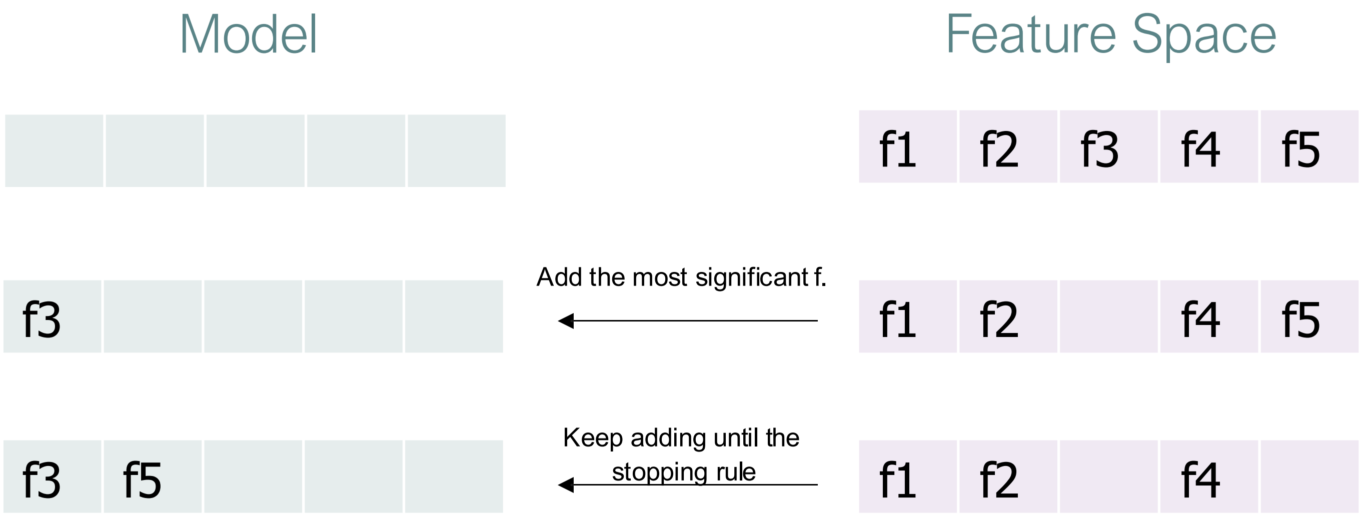
	Feat 1	Feat 2	Feat 3	Feat 4
Feat 1	1.00	-0.53	0.86	0.52
Feat 2	-0.53	1.00	-0.74	-0.61
Feat 3	0.86	-0.74	1.00	0.79
Feat 4	0.52	-0.61	0.79	1.00

Task Definition
Data Preparation
Model Training
Model Evaluation

Dimensionality Reduction: Feature Selection

Feature selection is essential for improving model accuracy and reducing complexity by selecting only the most relevant features. Here are the main methods:

- Forward Feature Selection



Model

Feature Space

f1 f2 f3 f4 f5

Add the most significant f.

f3 | f1 f2 f4 f5

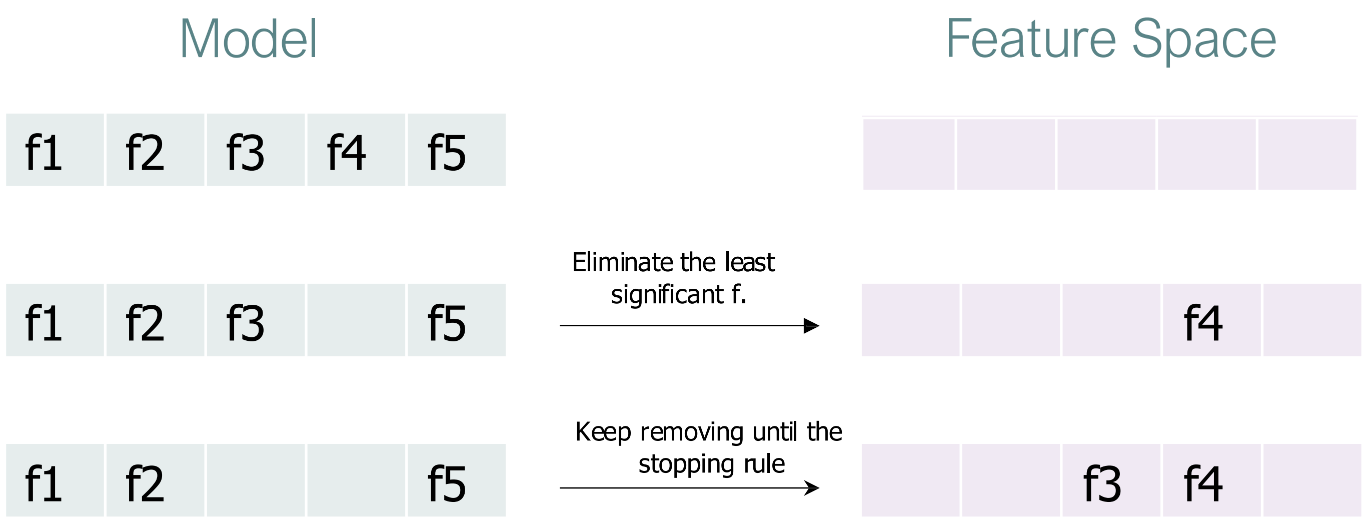
Keep adding until the stopping rule

f3 f5 | f1 f2 f4

Dimensionality Reduction: Feature Selection

Feature selection is essential for improving model accuracy and reducing complexity by selecting only the most relevant features. Here are the main methods:

- Backward Feature Selection



Model

Feature Space

f1 f2 f3 f4 f5

Eliminate the least significant f.

f1 f2 f3 f5 → f4

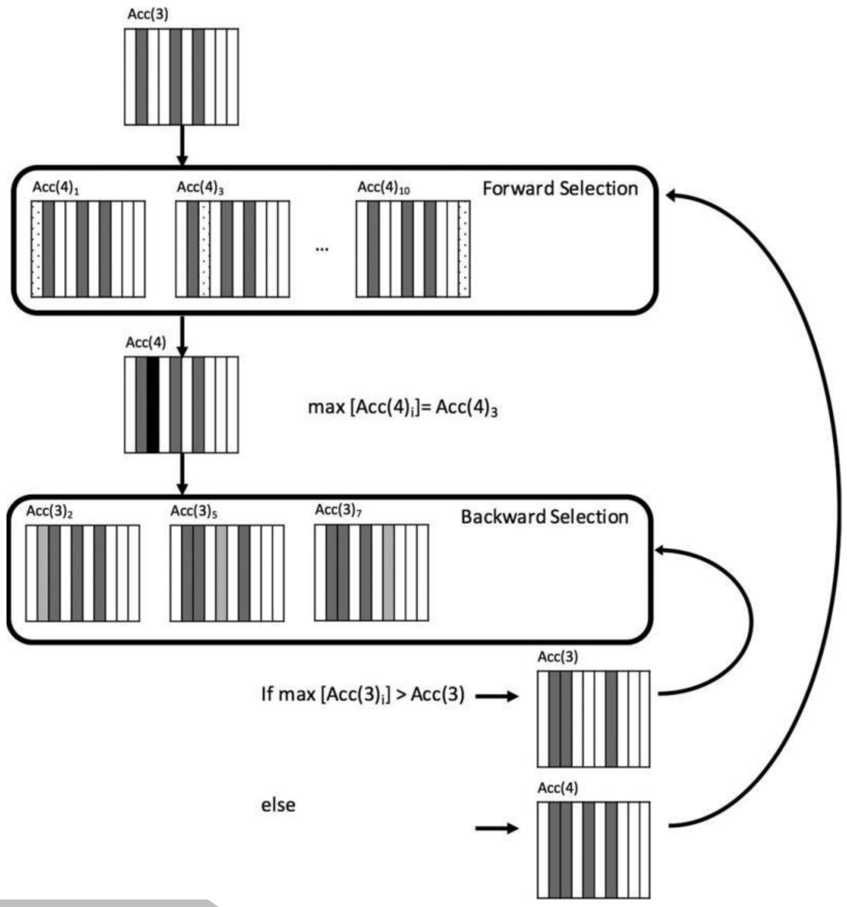
Keep removing until the stopping rule

f1 f2 f5 → f3 f4

Dimensionality Reduction: Feature Selection

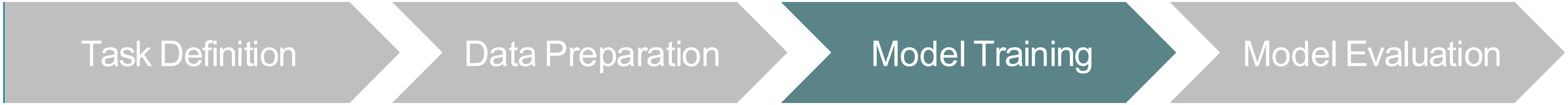
Feature selection is essential for improving model accuracy and reducing complexity by selecting only the most relevant features. Here are the main methods:

- Forward/Backward Feature Selection



Example

ML Model for Time Series/Signals Pipeline:



Task Definition

Data Preparation

Model Training

Model Evaluation

-Classification (binary, multi-label)

-Regression

-Handling abnormalities

-Data/Signal Processing

-Feature Extraction

-Dimensionality Reduction

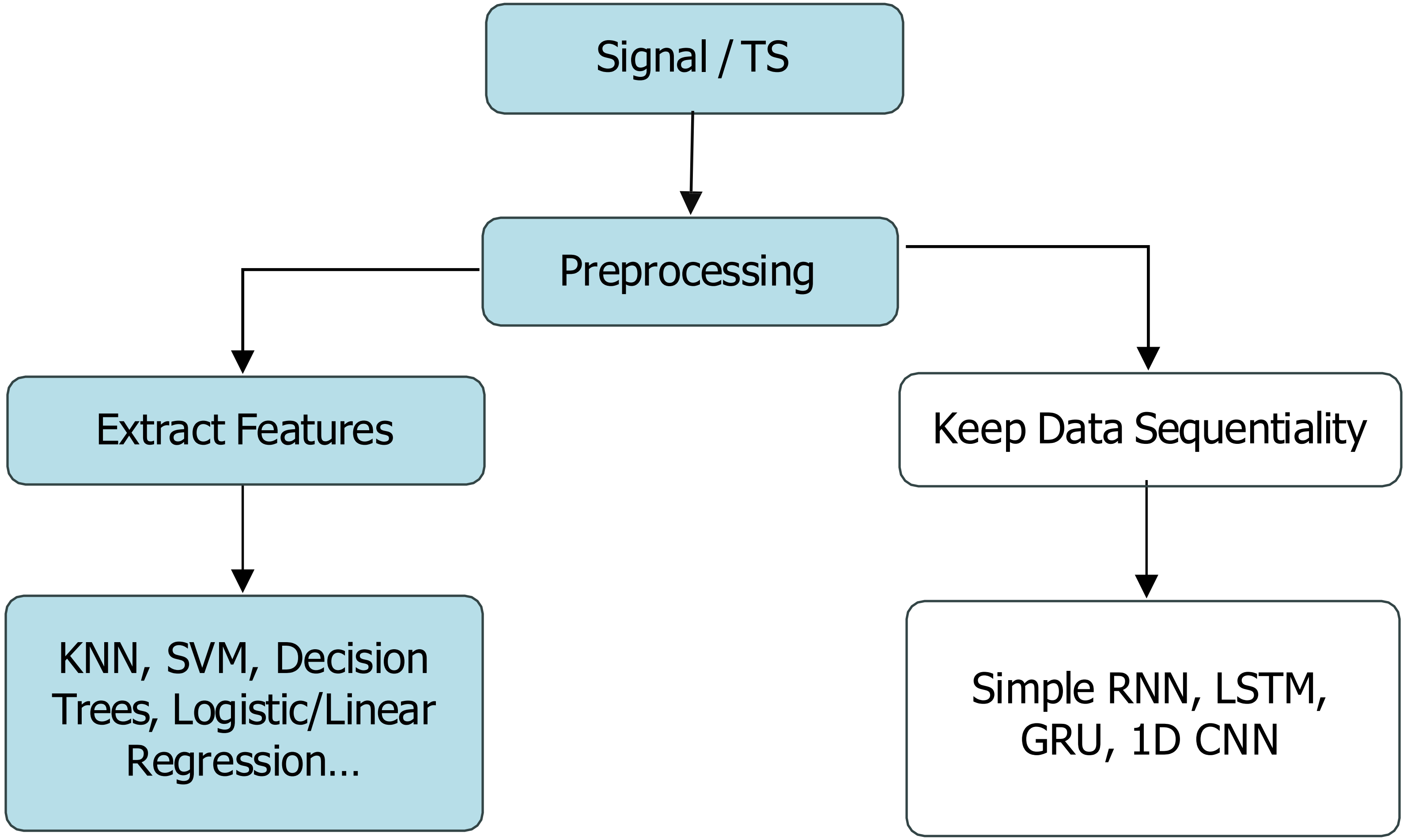
-Decision Trees, SVM, Regression…

-RNN, GRU, LSTM

- Train / Test split

-Cross-validation

-LOOCV



Signal / TS

Preprocessing

Extract Features

Keep Data Sequentiality

KNN, SVM, Decision Trees, Logistic/Linear Regression…

Simple RNN, LSTM, GRU, 1D CNN

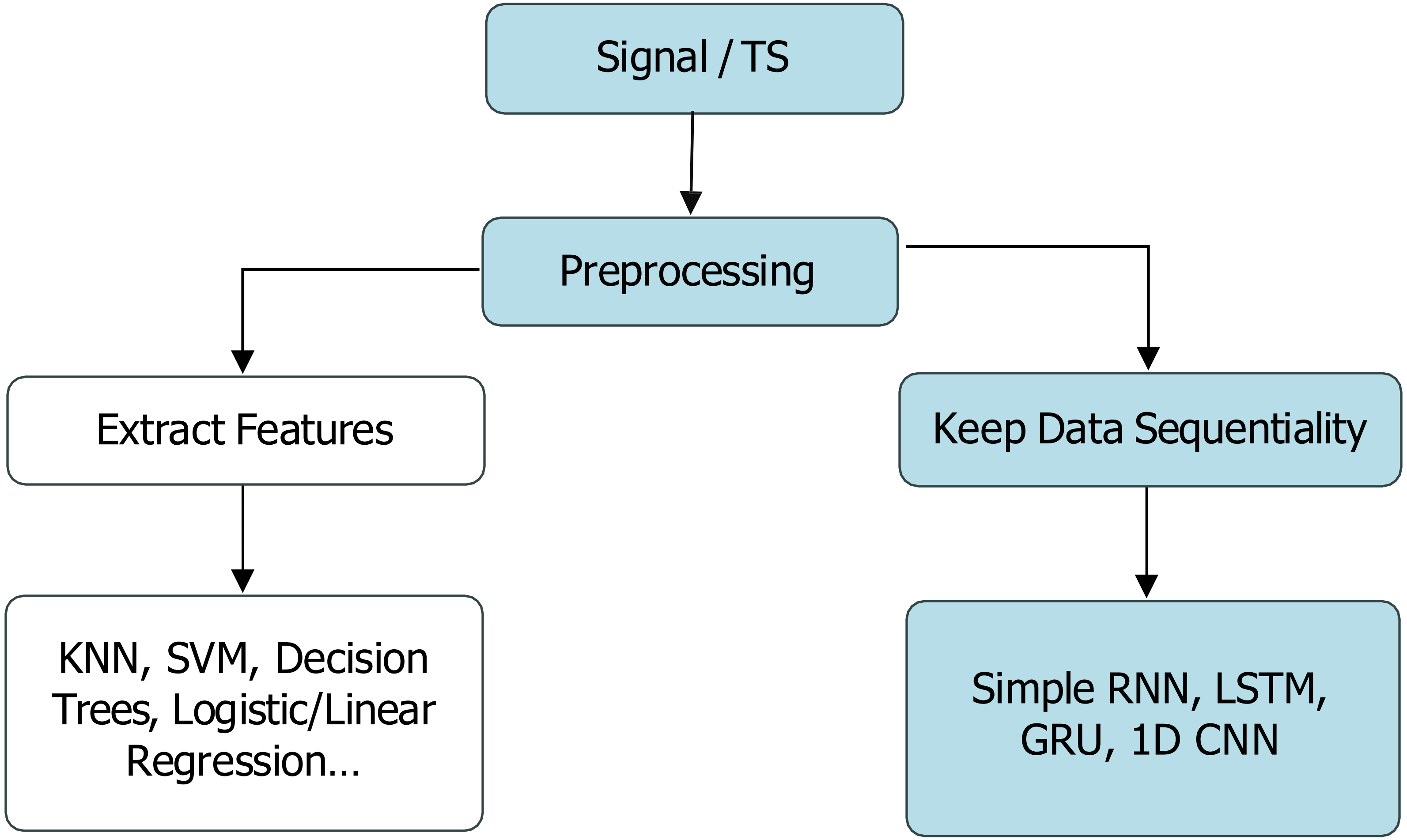
Most famous ML models that work with tabular data

Regression

- Linear regression, and all its regularized variants (Ridge regression, Lasso regression, ecc)
- Decision Tree based models and ensembles (Random Forests, Xgboost, ecc)
- K-Nearest-neighbours

Classificaition

- Logistic regression
- Decision Tree based models and ensembles (Random Forests, Xgboost, ecc)
- K-Nearest-neighbours
- Support Vector Machines
- Naive Bayes



Signal / TS

Preprocessing

Extract Features

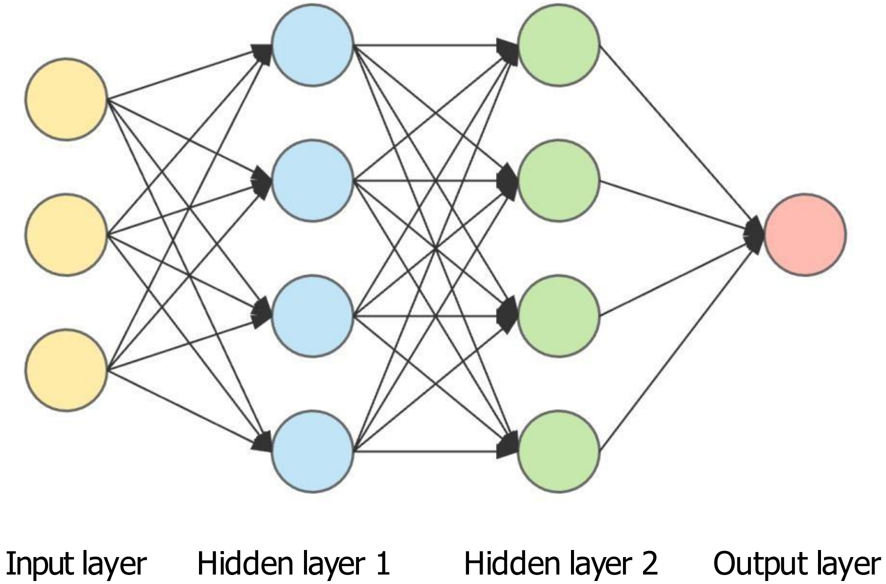
Keep Data Sequentiality

KNN, SVM, Decision Trees, Logistic/Linear Regression…

Simple RNN, LSTM, GRU, 1D CNN

Model training: Neural Networks

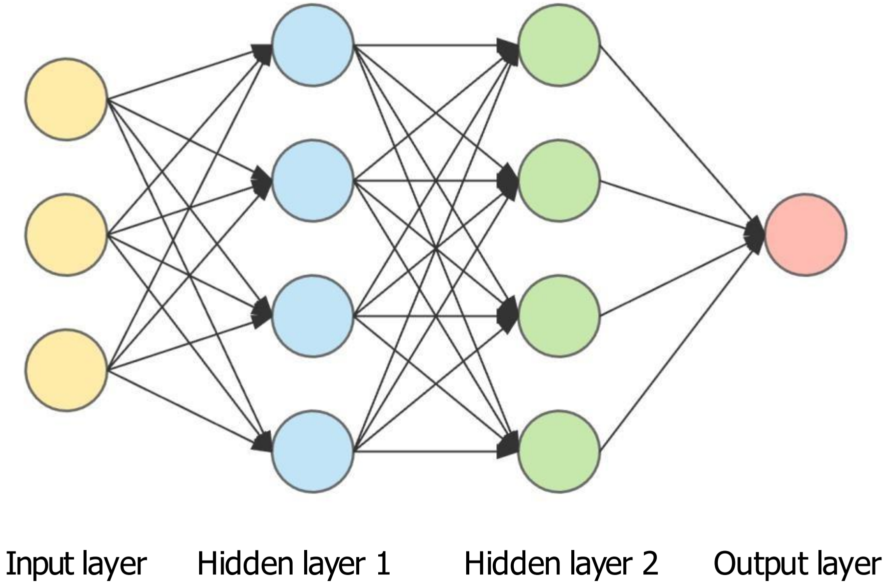
Neural networks are machine learning models inspired by the human brain, consisting of interconnected nodes (neurons) organized in layers.



Model training: Neural Networks

Neural networks are machine learning models inspired by the human brain, consisting of interconnected nodes (neurons) organized in layers.

Each neuron takes inputs, applies weights and biases, passes the result through an activation function, and outputs a value.

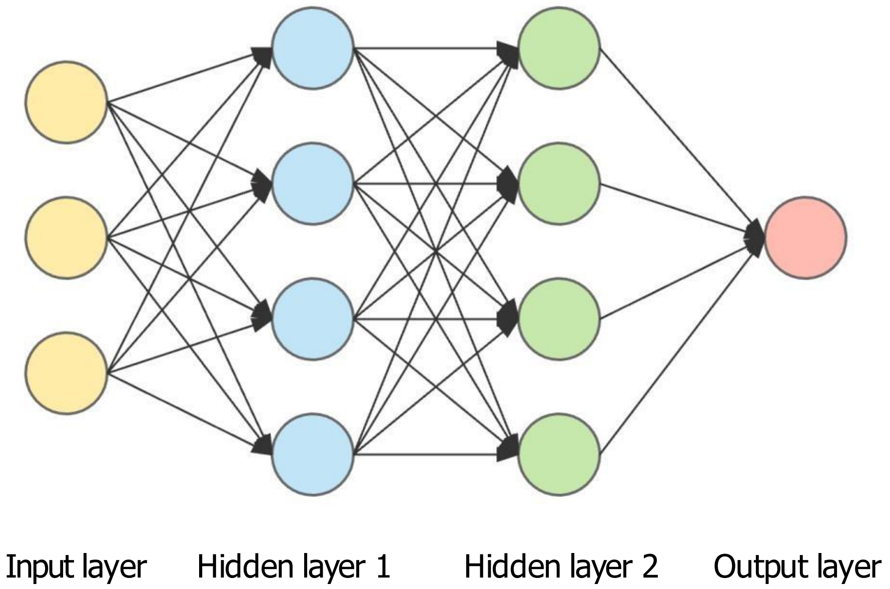


Model training: Neural Networks

Neural networks are machine learning models inspired by the human brain, consisting of interconnected nodes (neurons) organized in layers.

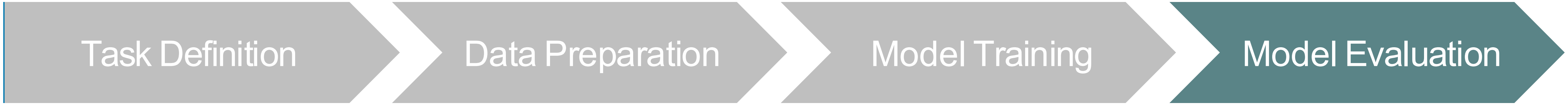
Each neuron takes inputs, applies weights and biases, passes the result through an activation function, and outputs a value.

The network learns by adjusting these weights and biases based on the error in predictions, which is minimized through backpropagation.



Task Definition → Data Preparation → **Model Training** → Model Evaluation

ML Model for Time Series/Signals Pipeline:


Task Definition

Data Preparation

Model Training

Model Evaluation

-Classification (binary, multi-label)

-Regression

-Handling abnormalities

-Data\Signal Processing

-Feature Extraction

-Dimensionality Reduction

-Decision Trees, SVM, Regression…

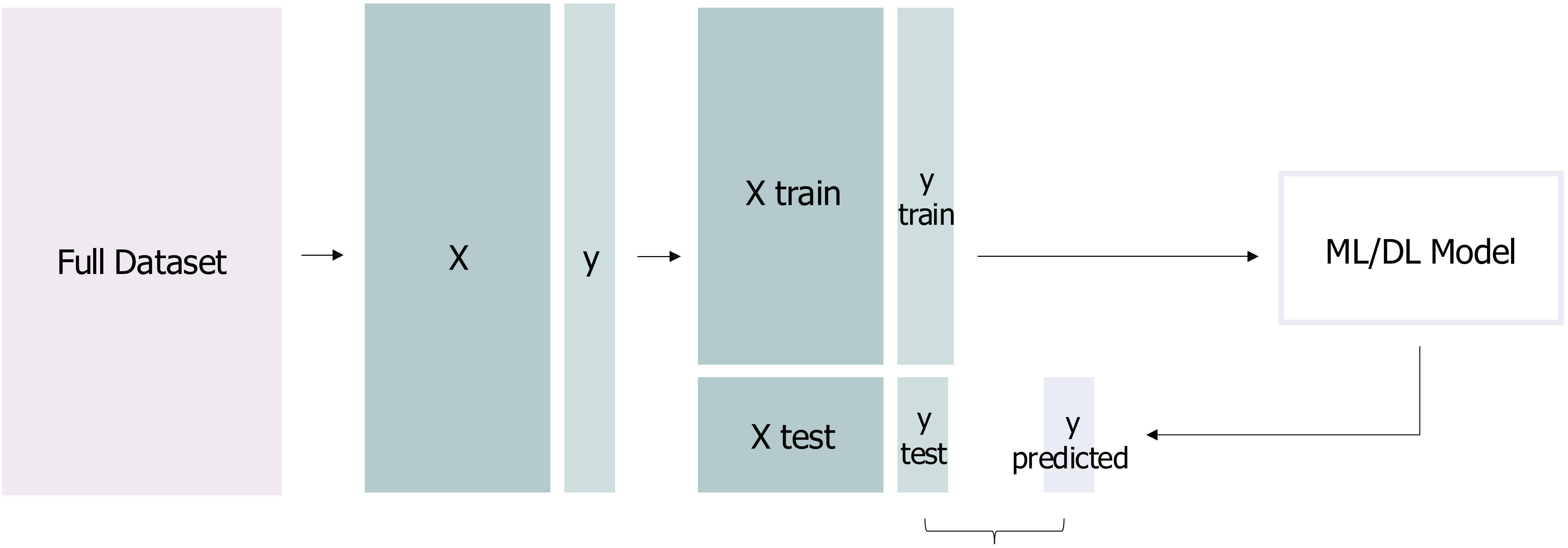
-RNN, GRU, LSTM

- Train / Test split

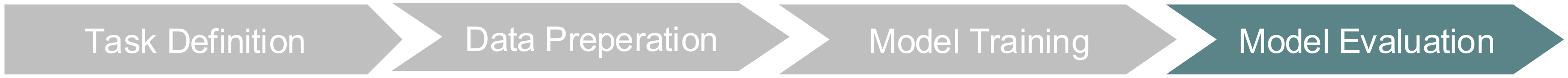
-Cross-validation

-LOOCV

Model Evaluation: Train-Test Split



Accuracy; Mean Absolute Error; Mean Squared Error…



ML Model for Time Series/Signals Pipeline:

Task Definition

-Classification (binary, multi-label)

-Regression

Data Preparation

-Handling abnormalities

-Data\Signal Processing

-Feature Extraction

-Dimensionality Reduction

Model Training

-Decision Trees, SVM, Regression…

-RNN, GRU, LSTM

Model Evaluation

- Train / Test split

-Cross-validation

-LOOCV

Model Evaluation: Overfitting

Overfitting occurs when a machine learning model learns not only the underlying patterns in the training data but also the noise and random fluctuations.

This happens when the model becomes overly complex, such as having **too many parameters relative to the amount of data**, enabling it to memorize the training data instead of generalizing to unseen data.

Task Definition

Data Preperation

Model Training

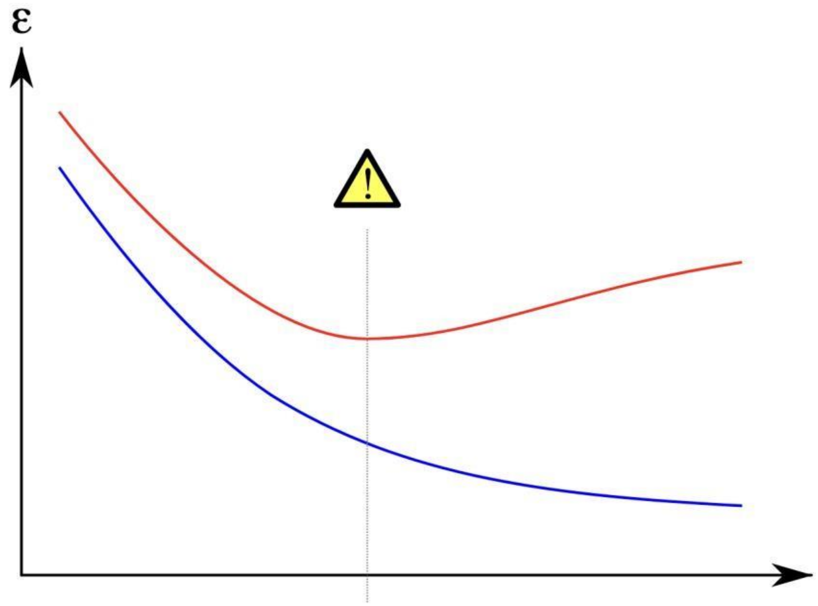
Model Evaluation

Model Evaluation: Overfitting

How to identify overfitting:

- Training vs. validation performance: The training accuracy is high, but validation accuracy is significantly lower

- Validation loss: The validation loss increases after a certain number of epochs, even as training loss continues to decrease.



Model Evaluation: How to prevent Overfitting

Regularization
Use dropout layers to randomly deactivate neurons during training.

Early stopping
Stop training when validation performance no longer improves.

Increase data
Gather more diverse training data or use data augmentation techniques.

Simplify the model
Reduce the number of parameters or use a less complex model.

Cross-validation
Use k-fold cross-validation to ensure the model generalizes well across subsets of the data

Model Evaluation: How to prevent Overfitting

Regularization
Use dropout layers to randomly deactivate neurons during training.

Early stopping
Stop training when validation performance no longer improves.

Increase data
Gather more diverse training data or use data augmentation techniques.

Simplify the model
Reduce the number of parameters or use a less complex model.

Cross-validation
Use k-fold cross-validation to ensure the model generalizes well across subsets of the data

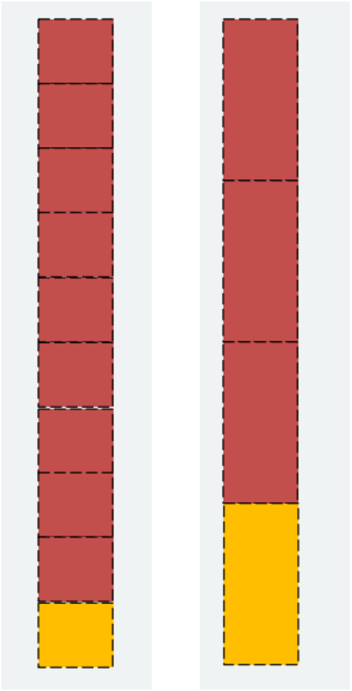
Model Evaluation when having a small dataset

Crossvalidation (k-fold)

Cross-validation is a more robust evaluation method where the dataset is split multiple times, creating different train-test splits.

The most common form is k-fold cross-validation, where the data is divided into k subsets (folds).

The model is trained k times, each time using a different fold as the test set and the remaining k-1 folds as the training set.



Task Definition → Data Preperation → Model Training → Model Evaluation

Model Evaluation when having a small dataset

Extreme Case: Leave-One-Out Crossvalidation

For a dataset with N data points, LOOCV creates N folds.

In each iteration, one data point is used as the test set, and the remaining N-1 points are used as the training set. The model is trained on the N-1 training points and tested on the single data point left out.

This process is repeated N times, with each data point serving once as the test set. The final performance metric is the average of the N individual test results