Scaling:

Normalization (MinMaxScaler)

Maps features to [0, 1]
keeps shape but compresses outliers. use when you need bounded inputs. when absolute range matters.

Standardization (Standard Scaler)
→ transforms features to zero mean and unit variance
→ often preferred for neural nets because it centers and scales inputs so gradient descent behaves better.

walk forward:

fit only on training data to avoid data leakage.

when doing walk forwad you must fit the scaler using only the current training window before transforming training inputs and the single forecast input. if you fit it on the entire series, future information leaks into training.

inverse transform for evaluation:

Always invert predictions back to the original scale before computing the RMSE, MAE, MAPE are any economic metric. for returns you may want to reapply transformations to get price forecasts.

Multivariate features:

As we have the multiple features then we need to Standarize each feature individually.

walk forward Validation

two main patterns.

1. Expanding window (growing training set)

→ Start with train 1.....T1o Predict To+1
Next fold train till To+1 to predict To+2
=> uses all ovilable prior information each Step
=> As the trainset ↑es can become expensive.

2. Rolling window (fixed window size)

→ train on window t.....t+w-1 to predict t+w
then shift window forward.
=> Keeps training size fixed, adapts to recent regime chan
=> loses older information needs careful window size tuning.

* Scaler Leakage fitting scaler on the whole dataset

* Lookahead features don't use columns but using future info.

* insufficient training examples relative to lookback

* Model reinitialization Vs warm start. reinitializing each step is safer but expensive. warm-starting keep weights can be efficient but can leak information indirectly if not tune on future validation carefully.

Choosing initial training size.

⇒ Should be large enough to allow the model to learn stable patterns. daily data 200+ may be minimal for high frequency instruments to adjust accordingly.

Sequence building and Lookback.

⇒ Lookback controls how many past time steps the model sees. typical values 10 — 60 daily data depending on pattern length. test multiple values.

Batching:

→ use shuffle = false for training time series if you want each batch to preserve contiguous sequences.

⇒ Alternatively use randomized mini batches that respect sequence boundaries

Hidden states & statefulness:

⇒ Pytorch, defaults to stateless LSTM (hidden state is not preserved across forward calls unless you manually keep it. for walk forward training it usual to treat each sequence independently.

⇒ if you use stateful LSTM, then must carefully reset hidden states between epochs and between training vs validation

Regularization & optimization

-> we can use dropout in LSTM if overfitting
=> Monitor training loss and validation loss

Warm-start vs full retrain:

-> full retrain each fold safer and avoids cumulative bias but expensive to run
-> fine tune but to ensure hyperparameters and validation are not tuned on future data.

Evaluation Metrics.

RMSE/MAE: measure magnitude of error
MAPE: Problematic when actuals near zero.

Feature engineering for time series:

=> Lagged returns
=> rolling statistics, rolling mean, rolling std, Skewness

using a coarser re-training frequency then still make daily prediction using the latest model.
using a rolling window with warm start
Limit model complexity for frequent retraining
parallelize folds where possible.