Normalization Timing in Machine Learning Pipelines: Before or After Train-Test Split

Data normalization, or feature scaling, is a critical preprocessing step in machine learning to ensure features contribute equally to model training, particularly for distance-based or gradient-descent algorithms. The key decision is whether to normalize before or after splitting data into training and test sets, as this impacts model generalization and validity.[1][2][3]

Recommended Practice

Normalize **after splitting**: Fit the scaler (e.g., StandardScaler or MinMaxScaler) exclusively on the training set, then apply the same transformation parameters (mean, std, min, max) to both training and test sets. This mimics real-world deployment where test data represents unseen future data. Scikit-learn pipelines enforce this via `fit(X_train)` followed by `transform(X_test)` .[3][4]

Normalization After Splitting

Advantages

- Prevents **data leakage**: Test set statistics (e.g., min/max values) do not influence training scaler parameters, yielding unbiased performance estimates.[2][5][1]
- Ensures **realistic evaluation**: Test set is transformed using only training statistics, simulating production where new data lacks full dataset knowledge.[1][3]
- Standard in libraries like scikit-learn, supporting cross-validation without leakage .

Disadvantages

- Slightly **less information** for training: Training scaler ignores test data, potentially yielding marginally suboptimal models in finite samples.[6]
- **Implementation complexity**: Requires separate fit/transform steps or pipelines, risking errors if misapplied.[6]

Normalization Before Splitting

Advantages

- **Simpler workflow**: Apply scaler to entire dataset once, then split—easier for quick prototyping.[6]
- **More training information**: Uses global statistics, potentially improving model fit slightly on small datasets.[6]

Disadvantages

- **Data leakage risk**: Test set influences scaler (e.g., global min/max), inflating performance metrics and hiding poor generalization.[7][5][2]
- **Unrealistic evaluation**: Overly optimistic test scores, as production data would use training-only parameters.[1]
- Violates ML best practices, leading to irreproducible results.[7]

Approach	Data Leakage	Model Performance Estimate	Implementation Ease	Best For
After Split [3]	None	Realistic	Moderate (pipelines)	Production models
Before Split [6]	High	Optimistic (biased)	Easy	Quick experiments only

In summary, normalize after splitting for robust, generalizable models, as endorsed by scikit-learn documentation and academic consensus. Tree-based models (e.g., Random Forest) are scale-invariant, reducing normalization needs.[3][7][1]

⁂

1. https://technicqa.com/should-i-normalize-before-or-after-test-train-split/
2. https://www.reddit.com/r/learnmachinelearning/comments/vjoo1r/should_you_scale_your_features_before_the/?tl=ru
3. https://sklearn.org/stable/modules/preprocessing.html
4. https://stackoverflow.com/questions/49444262/normalize-data-before-or-after-split-of-training-and-testing-data
5. https://www.reddit.com/r/learnmachinelearning/comments/j5k304/preprocessing_before_or_after_data_split_feature/
6. https://jamesmccaffreyblog.com/2020/05/27/should-you-normalize-and-encode-data-before-train-test-splitting-or-after-splitting/

7. https://arxiv.org/html/2506.08274v2
8. https://stackoverflow.com/questions/63181857/is-it-best-to-apply-minmaxscaler-to-your-dataset-before-splitting-into-training
9. https://community.altair.com/discussion/50465/normalising-data-before-data-split-or-after
10. https://www.youtube.com/watch?v=lCK9PSyHjno
11. https://www.reddit.com/r/datascience/comments/ae8u83/scaling_data_step_before_or_after_the_training/
12. https://www.reddit.com/r/AskStatistics/comments/yp65c8/when_scalingnormalizing_data_prior_to_applying_a/
13. https://www.reddit.com/r/MLQuestions/comments/1jzwqt5/is_normalizing_before_traintest_split_a_data/
14. https://pmc.ncbi.nlm.nih.gov/articles/PMC11978981/
15. https://pmc.ncbi.nlm.nih.gov/articles/PMC11419616/
16. https://www.oreateai.com/blog/data-preprocessing-in-machine-learning-detailed-explanation-of-dataset-splitting-and-normalization-methods/d5f39700f925783588228a7d5f465495
17. https://pmc.ncbi.nlm.nih.gov/articles/PMC12383348/
18. https://scikit-learn.org/stable/modules/preprocessing.html
19. https://www.datacamp.com/tutorial/normalization-in-machine-learning
20. https://www.reddit.com/r/MachineLearning/comments/utfuzj/discussion_data_preprocessing_before_or_after/