# Checkpoint 3 Guide: Model Evaluation Plan

## Evaluation Plan

* **Data Splits and Leakage Concerns**
  + **Geographic leakage**
    - Nearby or correlated spatial units split across train/test.
    - Example: adjacent houses, stores in the same shopping center, patients from the same hospital.
    - Tools: manual grouping, [GroupKFold](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html)
      * GroupKFold alone is often insufficient: custom splitters or splitting pipelines (e.g. first use spatial clustering techniques to assign groups, then use GroupKFold).
  + **Temporal leakage**
    - Future information leaking into past predictions.
    - Example: using 2023 features to predict 2022 outcomes.
    - Tools: [TimeSeriesSplit](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html)
      * TimeSeriesSplit is often not the correct structure and you will need to write a custom splitter. For example, if data only becomes available one week before predictions are needed it is inappropriate to train on all data up to the prediction time.
  + **Group-level leakage**
    - Data about the same entity split across train/test.
    - Example: Suppose you want to predict how a new student will perform on a standardized test at a new school. Your rows are individual student test events. If you use train\_test\_split or KFold, students from the same school end up in both train and test sets, so you never measure generalization to unseen schools. The fix is to split at the school level.
    - Tools: [GroupKFold](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html), [GroupShuffleSplit](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupShuffleSplit.html), [LeaveOneGroupOut](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeaveOneGroupOut.html)
  + **Feature leakage**
    - Variables that encode target information directly (or are highly correlated).
    - Tools: [PermutationImportance](https://scikit-learn.org/stable/modules/permutation_importance.html) in sklearn, [feature\_selection](https://scikit-learn.org/stable/modules/feature_selection.html)
  + **Post-outcome leakage**
    - Including features that were only known after the prediction time.
    - No direct tool; requires domain awareness.
  + **Overfitting the split**
    - Repeatedly adjusting models or features to perform well on a single validation/test split risks overfitting that particular partition rather than learning patterns that generalize.
    - Mitigation: use multiple splits (e.g., cross-validation, nested CV). Always keep a truly final holdout set untouched until the very end.
    - You should only ever evaluate **one model** on the final test set: it isn't another validation set.
  + **IMPORTANT NOTE**
    - In many cases, the built-in splitters won’t match your problem. You may need to implement a custom splitting strategy that explicitly mirrors the situations where you expect the model to generalize. Think carefully about the real deployment setting, and design your split to reflect it.
* **Metrics and Objectives**
  + **Mismatch between metric and goal**
    - RMSE vs. tail risk, AUC vs. calibration.
    - Tools: [sklearn.metrics](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics)
  + **Class imbalance**
    - Accuracy may mislead; use precision/recall, F1, ROC/PR.
    - Tools: [classification\_report](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html), [confusion\_matrix](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)
  + **Calibration**
    - Are predicted probabilities trustworthy?
    - Tools: [calibration\_curve](https://scikit-learn.org/stable/modules/generated/sklearn.calibration.calibration_curve.html), [CalibratedClassifierCV](https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html)
  + **Decision costs**
    - Cost-sensitive metrics.
    - Tools: [make\_scorer](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.make_scorer.html) for custom metrics.
* **Robustness and Stress Testing**
  + **Sensitivity to outliers**
    - Check if outliers dominate performance.
    - Tools: [RobustScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html), [isolation\_forest](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html)
  + **Distribution shifts**
    - Test on different cohorts (time, location, demographics).
    - Tools: manual holdout sets, [partial\_dependence](https://scikit-learn.org/stable/modules/generated/sklearn.inspection.partial_dependence.html) for feature sensitivity.
  + **Adversarial leakage checks**
    - Drop suspect features, shuffle targets.
      * If your model fits shuffled targets just as well as the real targets, then you know that you either have data leakage or you are over-fitting to your training splits somehow.
* **Practical Pitfalls**
  + **Train/test contamination during preprocessing**
    - Fit scalers/imputers only on training data.
    - Tools: [Pipeline](https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html)
  + **Hyperparameter tuning leakage**
    - Need nested CV or separate validation.
    - Tools: [GridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html), [RandomizedSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html) with proper CV strategy, [Optuna](https://optuna.org/).
  + **Data snooping**
    - Avoid peeking at test data.
    - Tools: enforce strict data split discipline.
* **Documentation and Transparency**
  + **Clear definition of unit of analysis**
    - Row = what? Transaction, user, visit, day?
  + **Clear description of split strategy**
    - Exactly how were train/test partitions created?
  + **Audit trail**
    - Why is this evaluation representative of the intended deployment?
    - What assumptions might break?

## Deliverables

* **Written / Conceptual**
  + A clear **evaluation plan** document (evaluation\_plan.md) describing:
    - The unit of analysis and how it shapes data splits.
    - The chosen split strategy (time, group, geographic, or custom) and why it reflects deployment.
    - Planned stress tests (outlier sensitivity, distribution shift checks, adversarial leakage tests).
* **Code / Repo Artifacts**
  + **src/splits/ directory**:
    - Custom splitter implementations (if needed), with docstrings explaining intended generalization scenario.
    - (Optional) Unit tests ensuring that no overlap exists between train/test sets for the specified grouping.
  + **notebooks/metric\_evaluation.ipynb**:
    - Implements evaluation with multiple metrics (sklearn.metrics functions, calibration plots, cost-sensitive metrics if relevant).
    - Includes confusion matrices, classification reports, ROC/PR curves (for classification) or error distribution plots (for regression).
    - Stores results in a tabular form (e.g., CSV in results/).
  + **results/ folder**:
    - Intermediate CSV/JSON tables of cross-validation scores, metric breakdowns, calibration curves.
    - Each file must specify the split strategy and date generated in its filename or metadata.
  + **Stress test scripts** in src/eval/:
    - Functions to re-evaluate models under perturbed conditions (dropped features, shuffled targets, specific cohorts).
    - Produces log output or plots stored under results/stress\_tests/.
  + **Note**: At the time of this checkpoint, you only need to have run your baseline models through your evaluation framework. Further modeling work will be done later.