## Feature Selection

**Why Feature Selection Matters**

* Curse of dimensionality, overfitting, training speed, interpretability.

**Key Terminology**

* Features vs. attributes, relevance, redundancy, variance, target leakage.

**Three Classic Families of Methods**

* Filter methods (e.g., correlation, mutual information, variance threshold).
* Wrapper methods (e.g., forward/backward selection, RFE).
* Embedded methods (e.g., LASSO/L1, tree-based impurity pruning).

**Dimensionality Reduction vs. Feature Selection**

* Clarify the difference (e.g., PCA transforms vs. selecting existing features).
* When to prefer one over the other.

**Evaluation Strategies**

* Using cross-validation pipelines; nested CV to avoid biased performance estimates.
* Metrics to watch (accuracy, AUC, F1, training time).

**Real-World Pitfalls & Best Practices**

* Data leakage, class imbalance effects, stability of selected sets.
* Domain knowledge integration.

**Simple End-to-End Example**

* Show a quick scikit-learn pipeline choosing top k features and comparing baseline vs. selected-feature model.

# Ensemble Learning

**Why Ensembles?**

* Variance-reduction, bias-reduction, robustness, Kaggle dominance.
* Quick contrast with bagging & boosting to set the stage for stacking/blending.

**Key Terminology**

* Base (level-0) learner, meta (level-1) learner, folds, hold-out set, out-of-fold (OOF) predictions.

**Stacking: Core Workflow**

* K-fold OOF prediction generation.
* Training the meta-model on OOF data.
* Diagram of two-level stack.

**Blending: The Lightweight Cousin**

* Train/validation split instead of K-folds.
* Pros: speed & simplicity; Cons: more data-hungry, less stable.

**Design Choices & Hyperparameters**

* Selecting diverse base models (trees, linear, NN).
* Meta-model options (linear regression, GBM, NN).
* Number of folds, blending hold-out size, regularization.

**Evaluation & Leakage-Free Pipelines**

* Nested CV or separate test set.
* Why fitting scalers & encoders inside each fold matters.
* Use of sklearn's StackingClassifier/Regressor vs. manual pipelines.

**Common Pitfalls + Best Practices**

* Collinearity of base-model outputs, overfitting in small data, long training times.
* Tricks: heterogeneous feature subspaces, feature selection per base model, regularizing meta-learner.

**Mini End-to-End Demo Idea**

* Stack logistic regression, random forest, and gradient boosting; meta-model = elastic-net.
* Compare single best base learner vs. stacked ensemble on accuracy/F1 and discuss results.