Feature Selection

The three broad categories of feature selection methods—**Filter**, **Wrapper**, and **Embedded**—represent different strategies for identifying the most predictive and least redundant variables in a dataset. They primarily differ in **when** the features are evaluated (before, during, or after model training) and **how** they are evaluated (statistically or based on model performance).

1. Filter Methods

Filter methods evaluate the relevance of features **independently of any specific machine learning model**. They are used as a preprocessing step to quickly filter out irrelevant features based on their intrinsic characteristics.

- **How it Works**: Features are scored using **statistical measures** (like correlation, χ2 tests, or information gain) that assess the relationship between the feature and the target variable, or between the feature and other features (e.g., removing highly correlated inputs).
- **Pros**:
    - **Fast and Scalable**: They are computationally inexpensive, making them ideal for high-dimensional datasets.
    - **Model Agnostic**: The selected feature set is general and can be used with any model.
- **Cons**:
    - **Ignores Feature Interactions**: They typically evaluate features individually and do not account for how features might work together to improve model performance.
    - **Not Optimized for a Model**: The chosen subset is not optimized for the ultimate model being trained, which may lead to suboptimal performance.
- **Examples**: Constant/Quasi-Constant Removal, Correlation, Variance Threshold, χ2 test, ANOVA F-test.

2. Wrapper Methods

Wrapper methods evaluate feature subsets by **training and testing a machine learning model** on each subset. The feature selection process "wraps" around the model, using its performance to guide the search for the optimal feature set.

- **How it Works**: A search algorithm (like greedy search) iteratively adds or removes features, and a model is trained on the resulting subset. The subset that results in the best model performance (measured by cross-validation) is selected.
- **Pros**:
    - **Model-Specific Optimization**: They tend to yield the best performing feature

    subset for the chosen model type.
  - **Captures Feature Interactions**: They inherently consider the interactions between features.
- **Cons**:
  - **Computationally Expensive**: Training a model many times for different feature subsets is very resource-intensive and slow, especially for large datasets.
  - **Risk of Overfitting**: The chosen features can sometimes be too well-tuned to the specific model and evaluation process.
- **Examples**: Forward Feature Selection, Backward Feature Elimination, Recursive Feature Elimination (RFE).

3. Embedded Methods

Embedded methods perform feature selection **during the model training process**. They combine the advantages of filter and wrapper methods by having the model building itself inherently penalize or select features.

- **How it Works**: The feature selection is built into the learning algorithm. The model's training objective function includes a mechanism (often regularization) that drives the coefficients or importance scores of irrelevant features to zero or near-zero.
- **Pros**:
  - **Balance of Speed and Accuracy**: They are faster than wrapper methods but typically more accurate than filter methods, as they interact with the model.
  - **Built-in Selection**: The selection process is a natural part of training, requiring no separate wrapper routine.
- **Cons**:
  - **Model Dependent**: The selection is specific to the algorithm being used (e.g., you can't use Lasso to select features for a basic Naive Bayes model).
- **Examples**: Lasso (L1) Regularization (which shrinks non-contributing coefficients to exactly zero), and **Feature Importance** derived from tree-based models (which rank features based on their contribution to impurity reduction).

Feature Selection Method	Relationship to Model	Evaluation Metric	Computational Cost	Result Quality
Filter	Independent (Pre-processing)	Statistical Scores (Correlation, χ2)	Low (Fastest)	Good (General feature set)
Wrapper	Dependent (Iterative Model Training)	Model Performance (Accuracy, F1 Score)	High (Slowest)	Highest (Optimized for the model)
Embedded	Dependent (During Model Training)	Model Coefficients/Feature Importance	Medium	High (Good balance of speed/accuracy)

Feature Selection Concepts

- **Constant-Quasi-Constant-Duplicates**: Techniques to remove features that provide little to no information:
  - **Constant**: A feature with only **one unique value** (i.e., all rows are the same).
  - **Quasi-Constant**: A feature where a single value appears in an **overwhelming majority** of the rows (e.g., 99% of the data).
  - **Duplicates**: Removing redundant columns that are **exact copies** of other columns.
- **Correlation**: A feature selection method where you **remove one of a pair of features** that are highly correlated with each other, as they provide redundant information to the model.
- **Filter-Statistical-Tests**: A category of feature selection (Filter Methods) that uses **statistical tests** (like χ2, ANOVA F-test, or information gain) to evaluate the relationship between each feature and the target variable, then keeps only the top-scoring features.

- **Filter-other-metrics**: Filter methods that rank features using metrics other than traditional statistical tests, such as **mutual information** or different measures of variance/spread.
- **Wrapper**: A category of feature selection methods that **evaluate subsets of features** by actually **training a model** on them. The selection process wraps around the model training (e.g., Forward Selection, Backward Elimination).
- **Embedded-linear-coefficients**: An approach where **linear models** (like Linear Regression or Logistic Regression) are used for feature selection. Features with **coefficients close to zero** are deemed less important and can be removed.
- **Embedded-Lasso**: A powerful type of embedded method that uses **Lasso (L1) regularization**. The L1 penalty forces the coefficients of the least important features to become **exactly zero**, effectively performing feature selection during the model training process.
- **Embedded-tree-importance**: An embedded method where **tree-based models** (like Random Forests or Gradient Boosting) are trained, and their **feature importance scores** (how much each feature contributes to reducing impurity) are used to rank and select the best features.
- **Hybrid-methods**: Techniques that **combine** different selection strategies (Filter, Wrapper, and Embedded methods) to achieve a more robust and effective feature subset. For example, using a Filter method to quickly eliminate a large number of bad features, followed by a more rigorous Wrapper method.