Feature Selection

Bruno Iochins Grisci

Generative AI Academy

Part 1: Feature Selection Algorithms

Overview

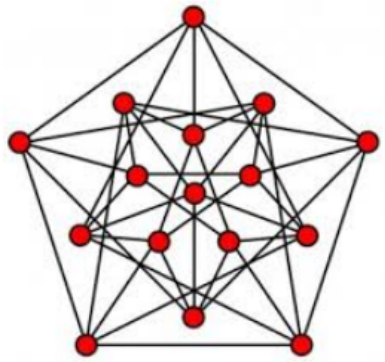
Who am I?

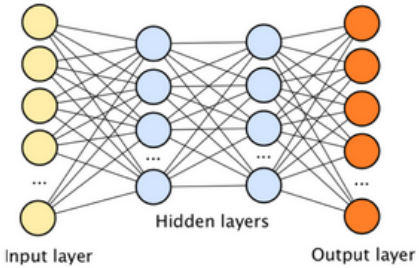
PhD in Computer Science
Professor at the Theoretical Informatics Department of UFRGS

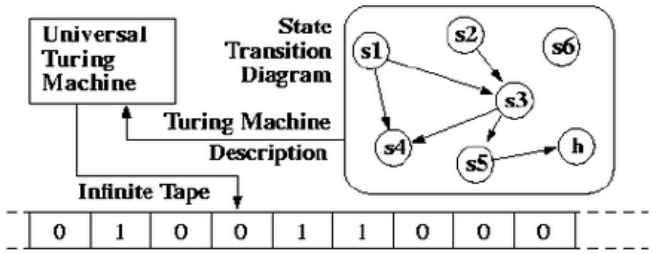
Teaching

Fundamentos de Algoritmos
Teoria dos Grafos e Análise Combinatória
Teoria da Computação
Inteligência Artificial na Educação

```
(define a 2)

(define emptylist '())

(define (gcd u v) ;defines a function
  (if (= v 0) u
      (gcd v (remainder u v))))

(cond((= a 0) 0) ;if (a==0) return 0;
     ((= a 1) 1) ;elsif(a==1)return 1;
     (else (/ 1 a))); else return 1/a;
```

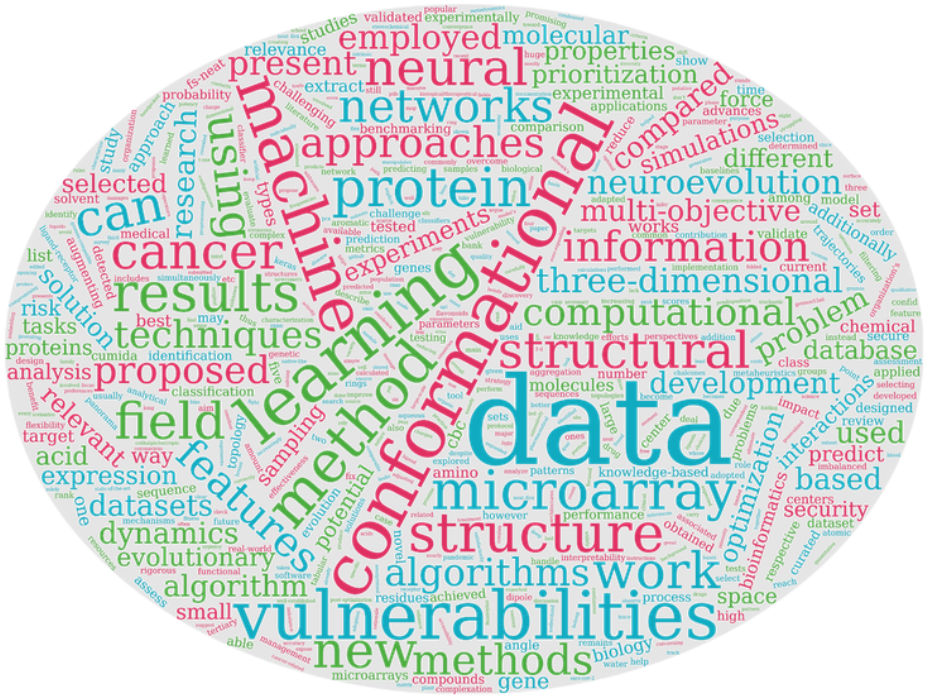






Research

Feature Selection
Deep Learning
Evolutionary Computation
Data Visualization
Bioinformatics
Explainable AI



Feature Selection

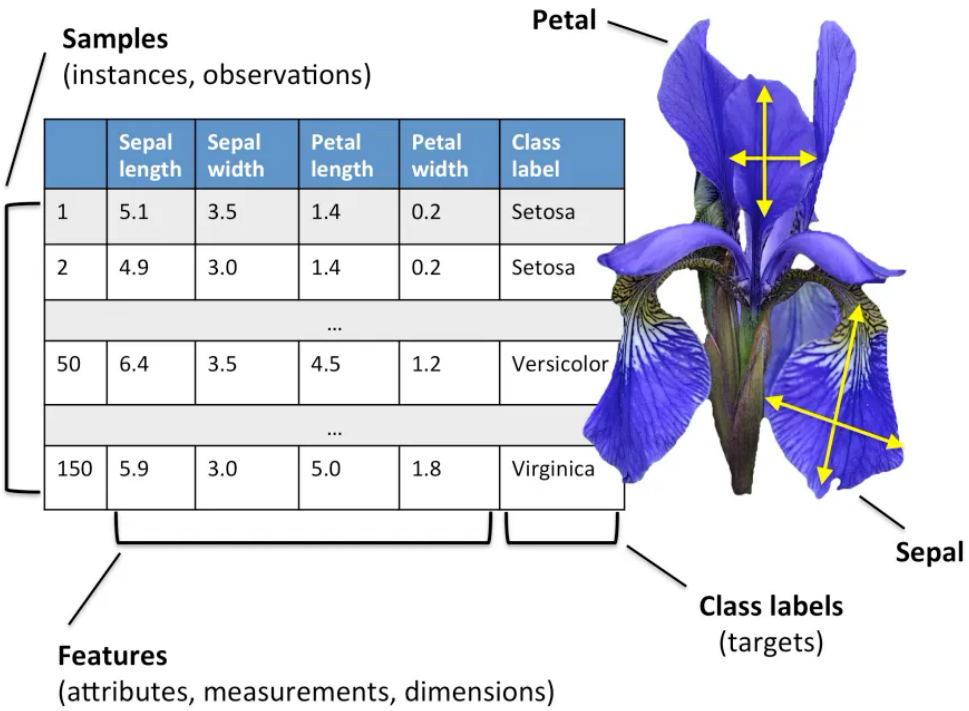
What is a Feature?

- A feature is a measurable property or attribute of a data sample.
- In tabular data, features are the columns (dimensions) of the dataset.
- Each sample (row) is represented as a feature vector.

Examples of Features

- Gene expression levels
- Pixel intensities in images
- Demographic variables in surveys

Features



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

https://vinlab.medium.com/mastering-machine-learning-with-scikit-learn-an-experiment-with-the-iris-dataset-4c649dc65acf

What is Feature Selection?

- Process of identifying a subset of relevant features from the original dataset.
- Aim: Improve model performance and interpretability.
- Maintains original feature semantics (unlike feature extraction).

Illustration of Feature Selection

Name	Employee ID	No. of year experience	Previous salary	Salary
Rahul	1	2	20000	40000
Aman	34	3	30000	50000
Ritika	31	5	50000	70000

Not useful Not useful

No. of year experience	Previous salary	Salary
2	20000	40000
3	30000	50000
5	50000	70000

https://www.shiksha.com/online-courses/articles/feature-selection-beginners-tutorial/

Dimensionality Reduction

Two main strategies:

Feature Selection

Selects a subset of existing features.

Feature Extraction

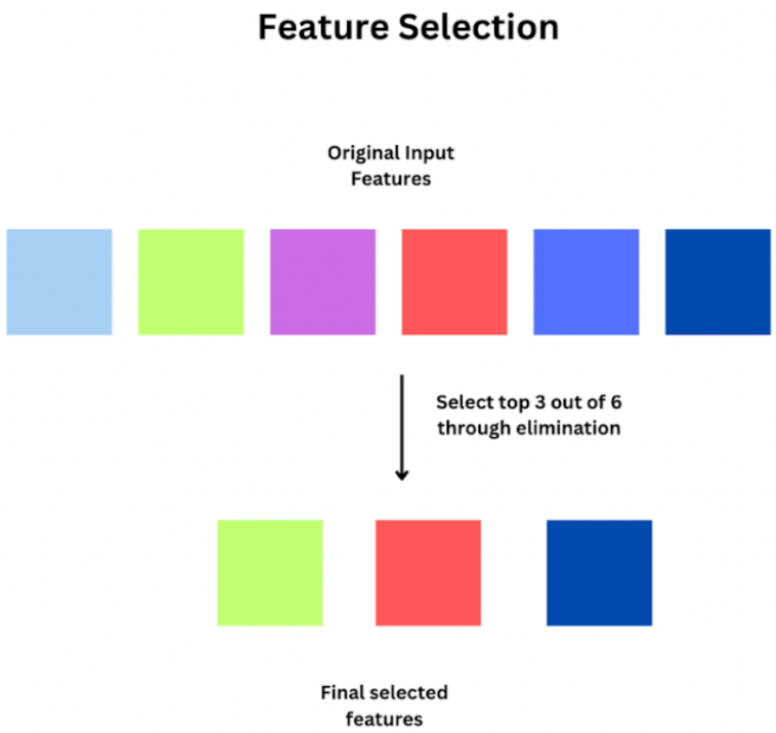
Transforms features into a new space (e.g., PCA).

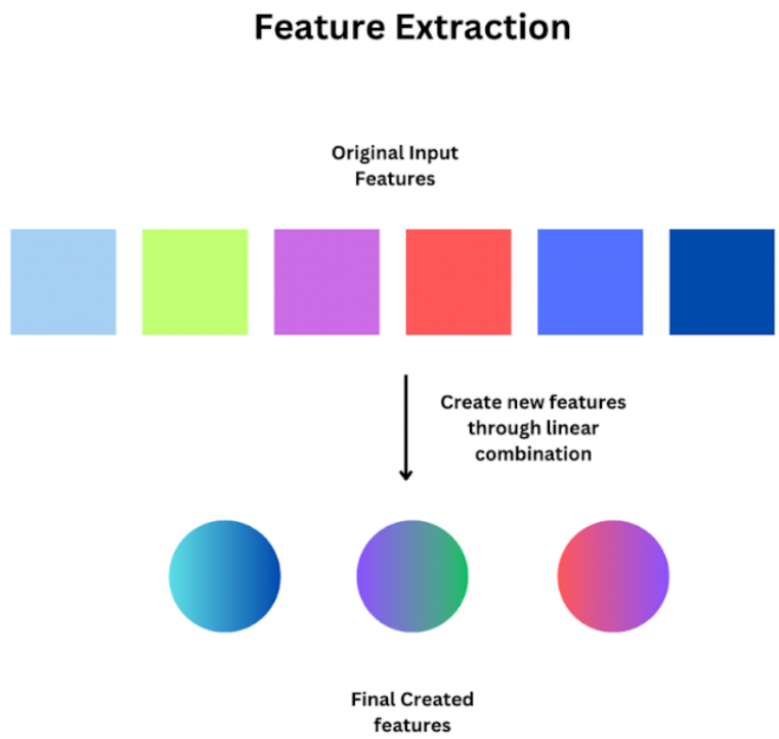
Selection vs Extraction

Comparison:

- **Selection:** Preserves interpretability
- **Extraction:** May improve performance but reduces interpretability

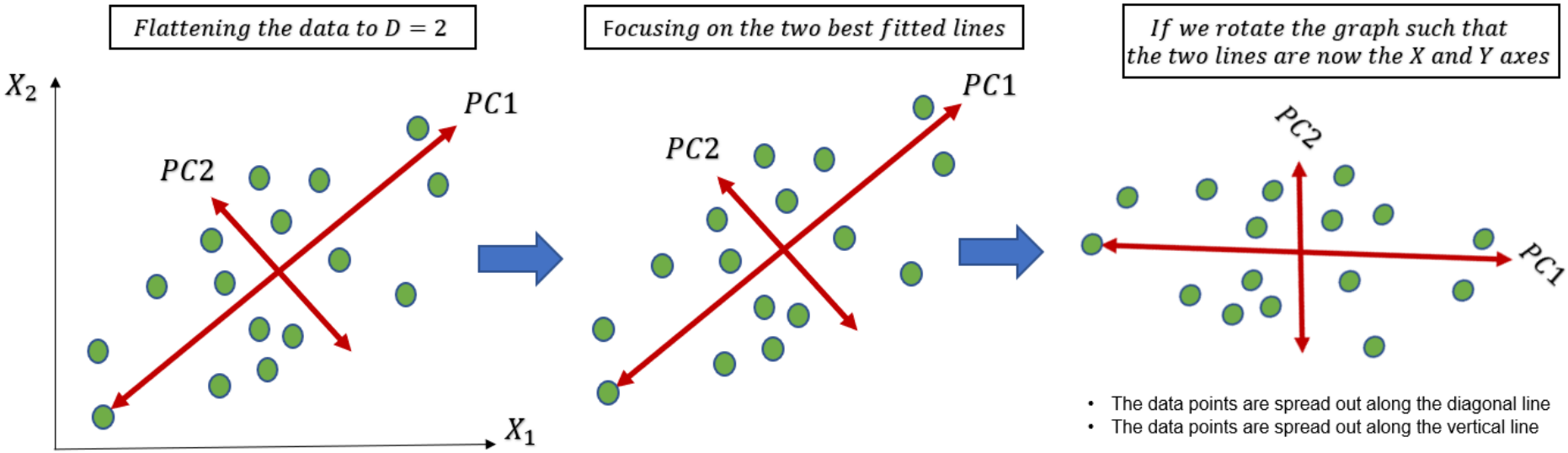
Selection vs Extraction





https://viso.ai/deep-learning/feature-extraction-in-python/

PCA



https://www.linkedin.com/pulse/gentle-introduction-principal-components-analysis-michael-wynn/

Why Feature Selection?

Key Motivations:

- **Accuracy:** Removes noise and overfitting risk
- **Memory:** Reduces model and data size
- **Interpretability:** Helps identify meaningful inputs

Performance and Dimensionality

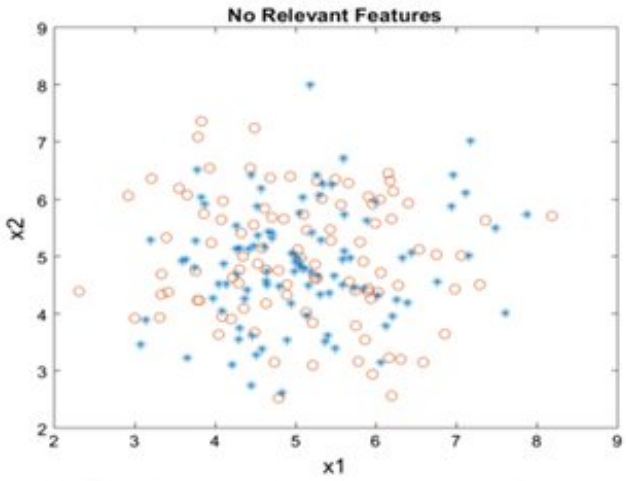
Useful in domains with high-dimensional data and limited samples (e.g., gene expression).

Relevant, Redundant, and Irrelevant Features

- **Relevant:** Contain information useful for prediction
- **Redundant:** Correlated with other features, add no new info
- **Irrelevant:** Unrelated to the target variable

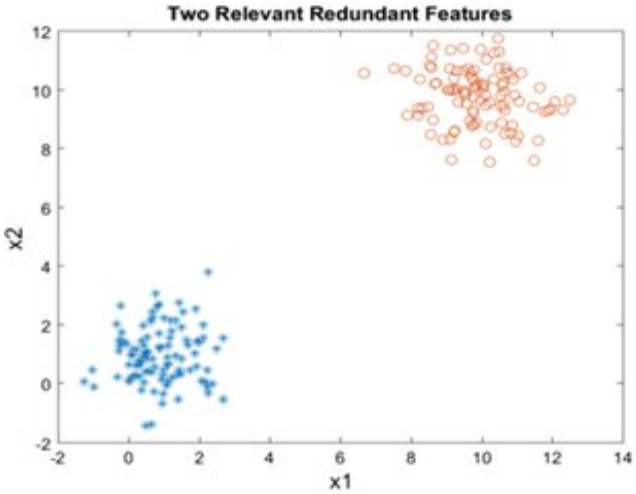
Goal of Feature Selection: Retain relevant, remove redundant/irrelevant.

Relevant, Redundant, and Irrelevant Features

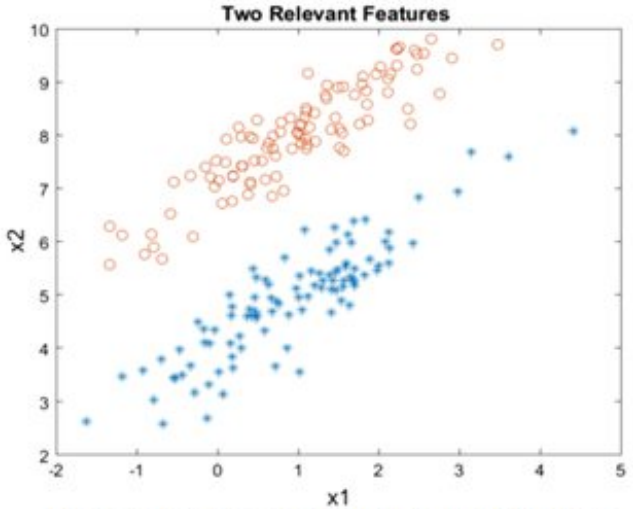


(a) Two-class example with no relevant features

(b) Two-class example with one relevant features



(c) Two-class example with two redundant features



(d) Two-class example with two relevant features

DOI: 10.1007/s11227-023-05758-3

How is Feature Selection Represented?

Three common forms:

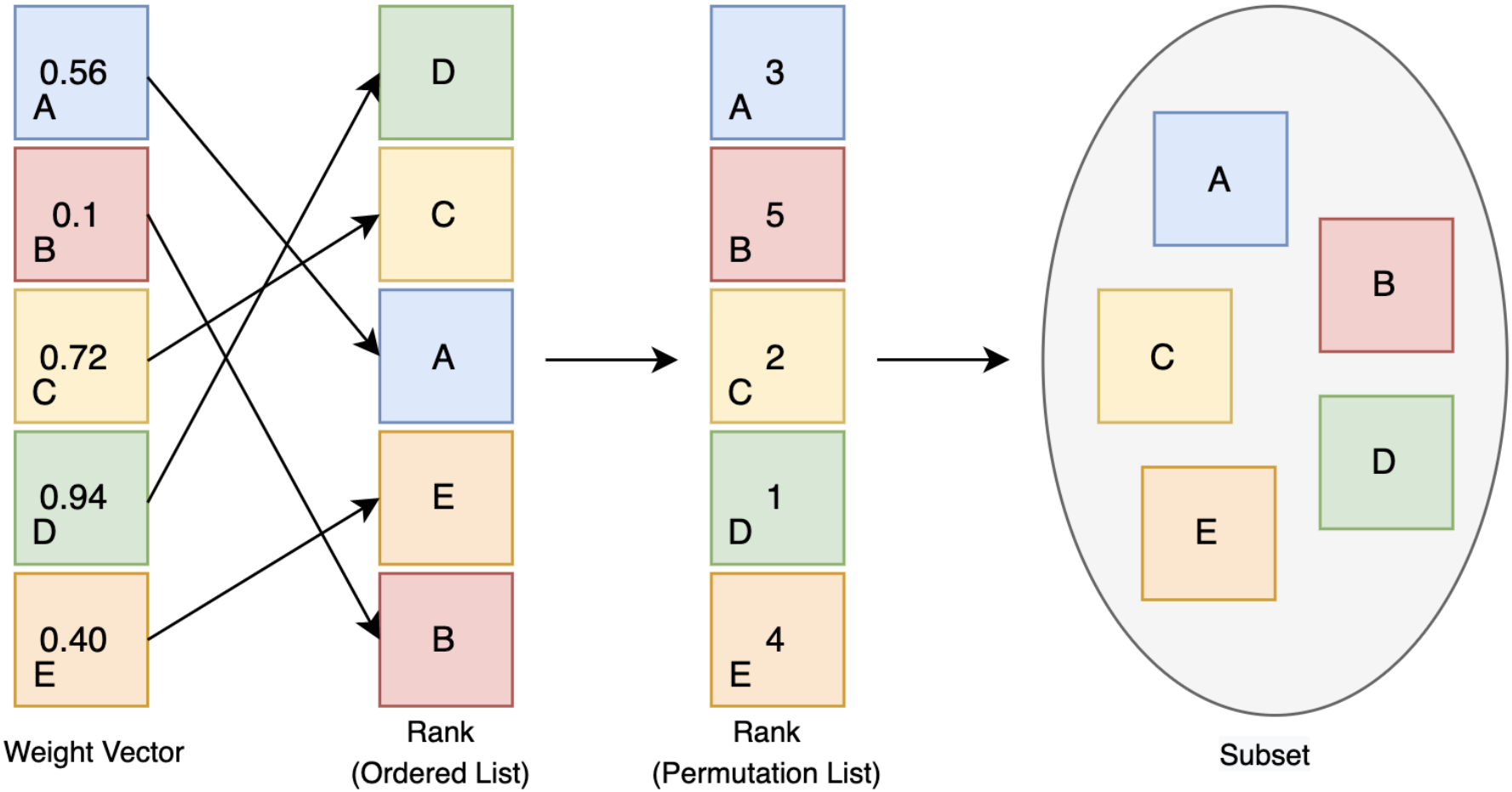
- **Weights:** Continuous importance values (e.g., 0.0 to 1.0)
- **Scores:** Ranking or statistical significance
- **Subsets:** Binary selection mask (0 or 1)

Representation Types: Comparison

Representation affects:

- Interpretability
- Integration with learning algorithms

Representation Types: Comparison



Theoretical Background

Theoretical Background

Foundations for Feature Selection:

- Statistics
- Information Theory
- Metaheuristics
- Supervised Learning

Statistics for Feature Selection

Statistical methods assess the relationship between features and class labels.

- These methods test whether feature distributions differ between classes.
- Important for high-dimensional, low-sample-size data.
- Especially common in filter-based methods.

Common Statistical Tests

- **t-test:** Assesses if two class means are significantly different.
- **ANOVA:** Generalizes t-test for multiple classes.
- **Kruskal-Wallis:** Non-parametric alternative to ANOVA.

Kruskal-Wallis Statistic:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)$$

- N = total number of samples
- k = number of classes
- n_i = number of samples in class i
- R_i = sum of ranks in class i

Information Theory for Feature Selection

Measures uncertainty and information contribution of features.

- Based on probability distributions.
- Common in filter methods due to efficiency and generality.

Key Concepts from Information Theory

Entropy:

$$H(X) = -\sum_{x \in X} P(x) \log P(x)$$

Mutual Information (MI):

$$I(X; Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

- Measures how much knowing X reduces uncertainty about Y
- Features with high MI to class labels are preferred

Metaheuristics for Feature Selection

Optimization techniques to search the space of feature subsets.

- Effective for complex, high-dimensional search spaces.
- Often used in wrappers and hybrids.
- Evaluate subsets using a fitness function (e.g., accuracy).

Popular Metaheuristics

- **Genetic Algorithms (GA)**: Inspired by natural selection.
- **Simulated Annealing**: Probabilistic exploration based on thermodynamics.
- **Particle Swarm Optimization (PSO)**: Models social behavior of swarms.

GA Example:

- Encode subset as binary chromosome.
- Use crossover, mutation, and selection.
- Fitness = performance on validation set.

Genetic algorithms

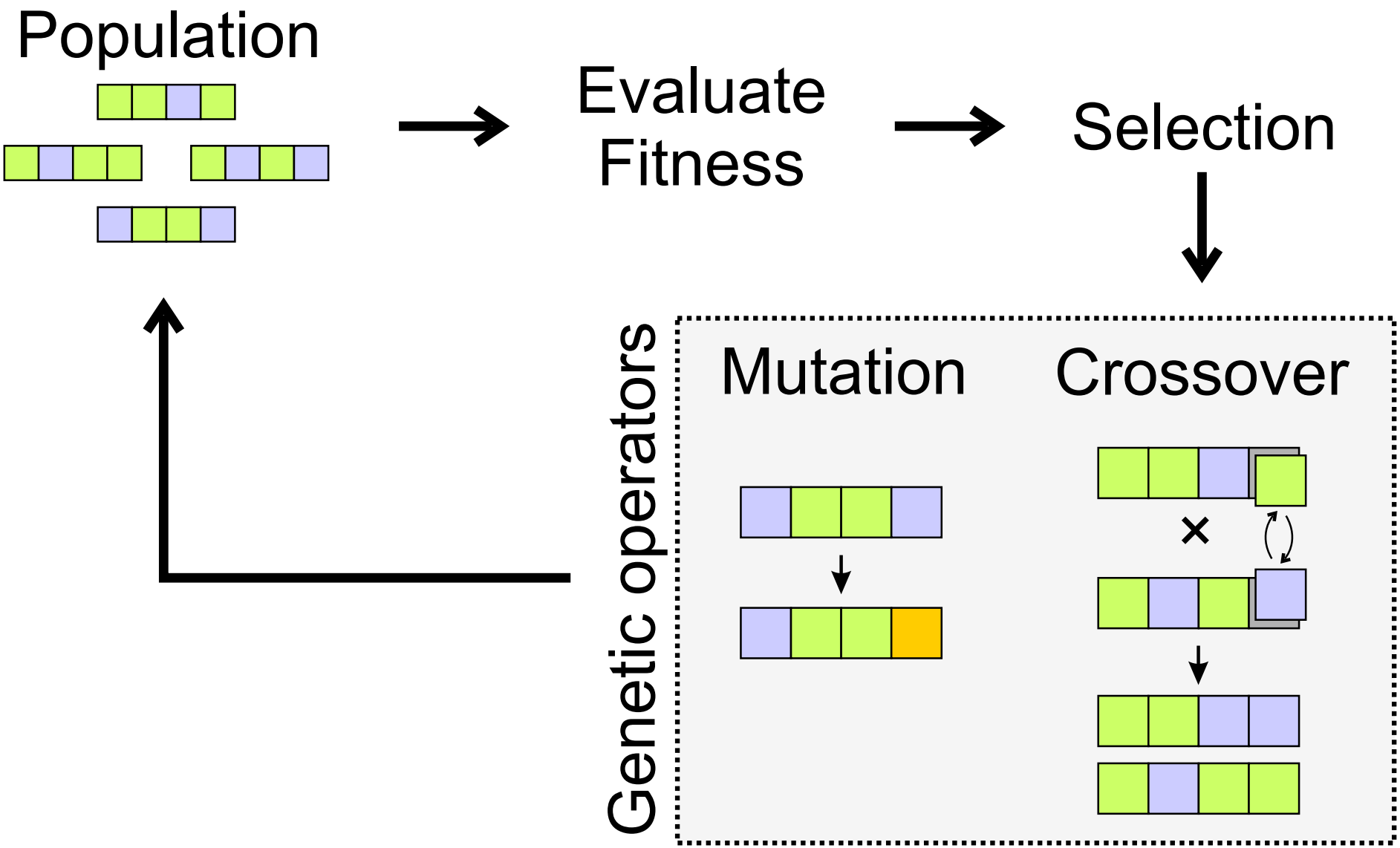
Algorithm 1 GA Pseudo-code

Inputs : **S**: The set of all attributes, **n_gen**: Maximum number of generations, **max_fitness**: Maximum fitness threshold

Output: **P**: The last population to be generated

```
1  P ← create_initial_population(S)
2  fitness ← evaluate_fitness(P)
3  for i ← 1 to n_gen or max(fitness) == max_fitness do
4  |   fittest ← selection(P, fitness)
5  |   offspring ← crossover(fittest)
6  |   offspring ← mutate(offspring)
7  |   elite_individuals ← elite(P)
8  |   P ← offspring + elite_individuals
9  |   fitness ← evaluate_fitness(P)
10 end
11 return P
```

Genetic algorithms



Supervised Learning and Feature Selection

Use model feedback to guide feature relevance.

- Embedded methods perform selection during training.
- Feature importance comes from model parameters.

Examples of Embedded Methods

- **Decision Trees:** Splits selected based on information gain.
- **LASSO:** ℓ_1 regularization sets irrelevant weights to zero.
- **SVM-RFE:** Recursive Feature Elimination based on SVM weights.

Properties

Properties of Feature Selection

Key Dimensions of FS Algorithms:

- Selection Strategy: Filter, Wrapper, Embedded
- Integration Mode: Hybrid, Ensemble
- Evaluation Scope: Univariate vs Multivariate
- Search Direction: Forward, Backward, or Bidirectional

Filter Methods

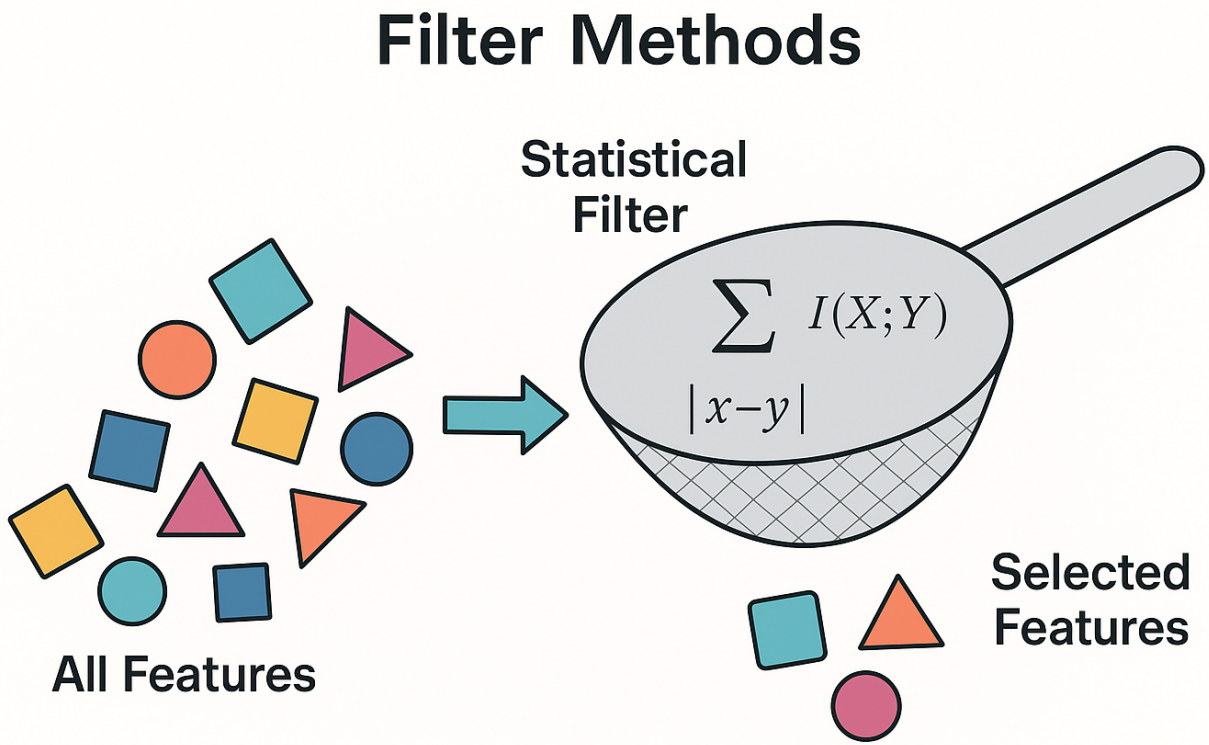
Select features based on intrinsic properties of data.

- Independent of learning algorithm.
- Fast, scalable, and generalizable.
- Use statistical, information-theoretic, or distance-based measures.

Examples:

- Kruskal-Wallis, Mutual Information, ReliefF

Filter Methods



Selection based on relevance scores – fast and model–agnosstic

Wrapper Methods

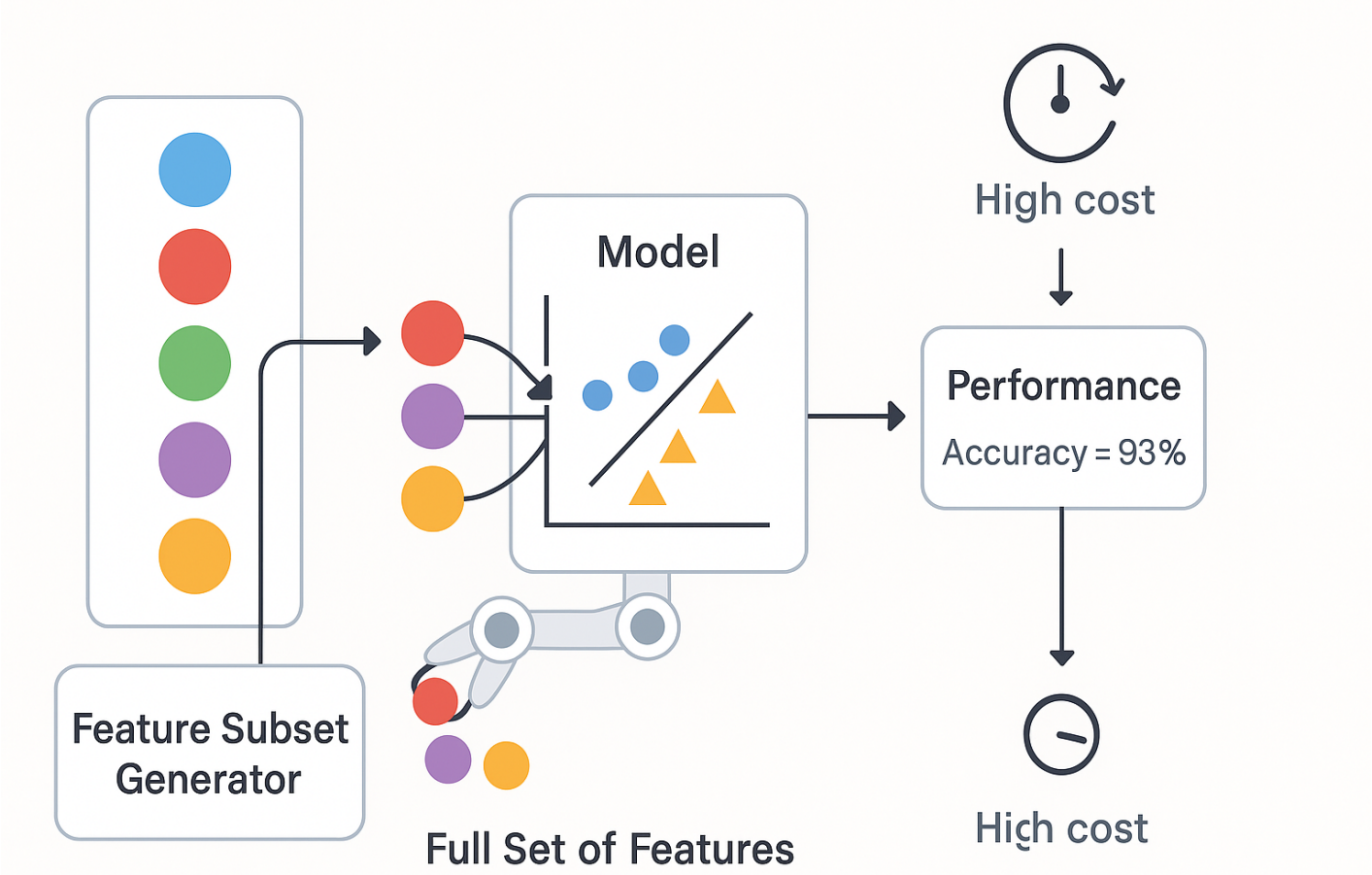
Use a learning algorithm to evaluate feature subsets.

- Train model on different subsets to assess performance.
- Higher accuracy but computationally expensive.
- Can overfit on small datasets.

Examples:

- SVM-RFE, Forward/Backward selection

Wrapper Methods



Embedded Methods

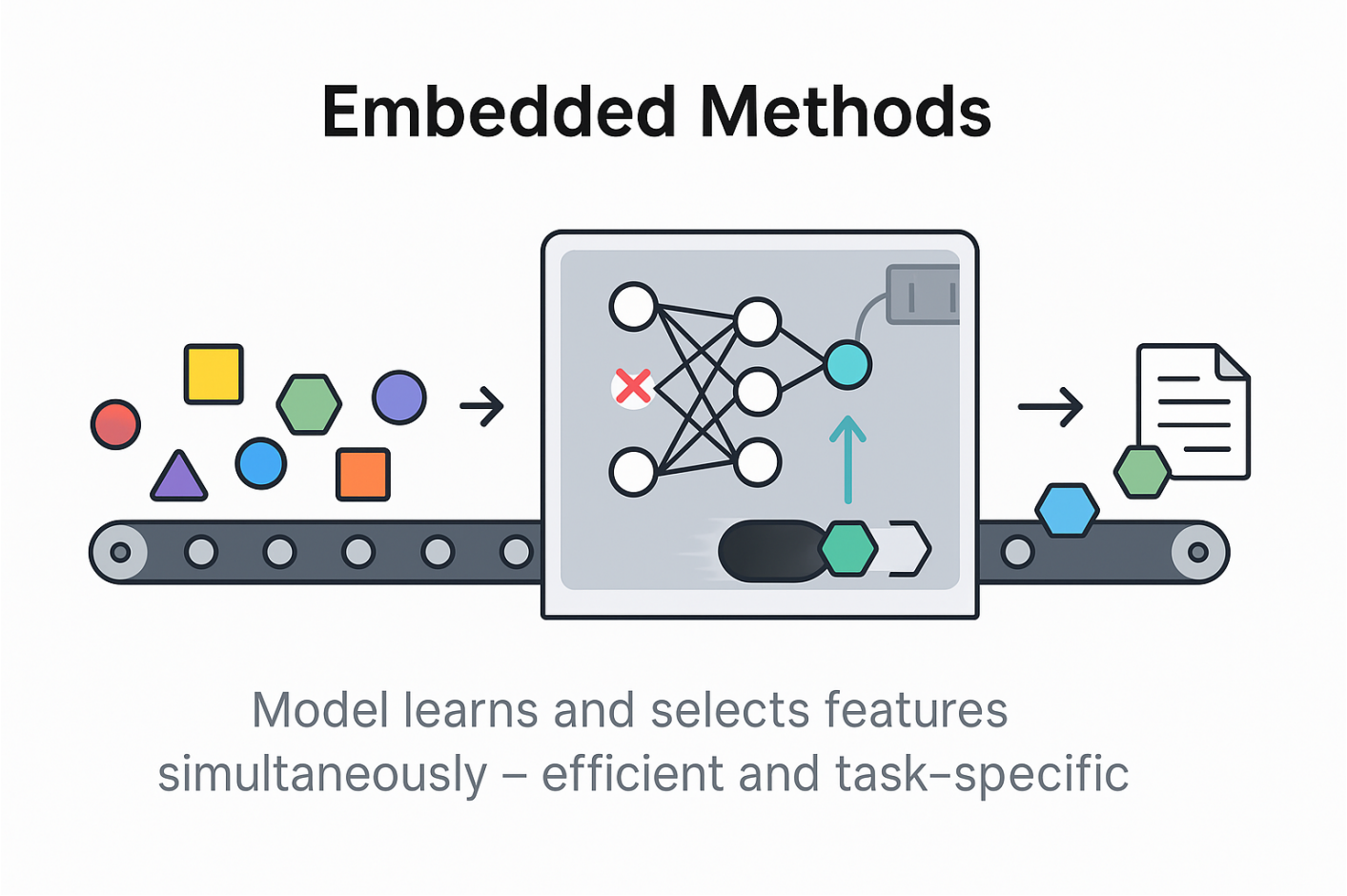
Perform selection as part of model training.

- Combine learning and selection steps.
- Efficient and task-specific.

Examples:

- LASSO, Decision Trees, Regularized SVMs

Embedded Methods



Hybrid and Ensemble Methods

Hybrid:

- Combine filter and wrapper stages.
- Leverage speed of filters and accuracy of wrappers.

Ensemble:

- Combine outputs of multiple FS algorithms.
- Improve stability and robustness.

Examples:

- Hybrid GA + filter, Ensemble based on bootstrapping

Univariate vs Multivariate

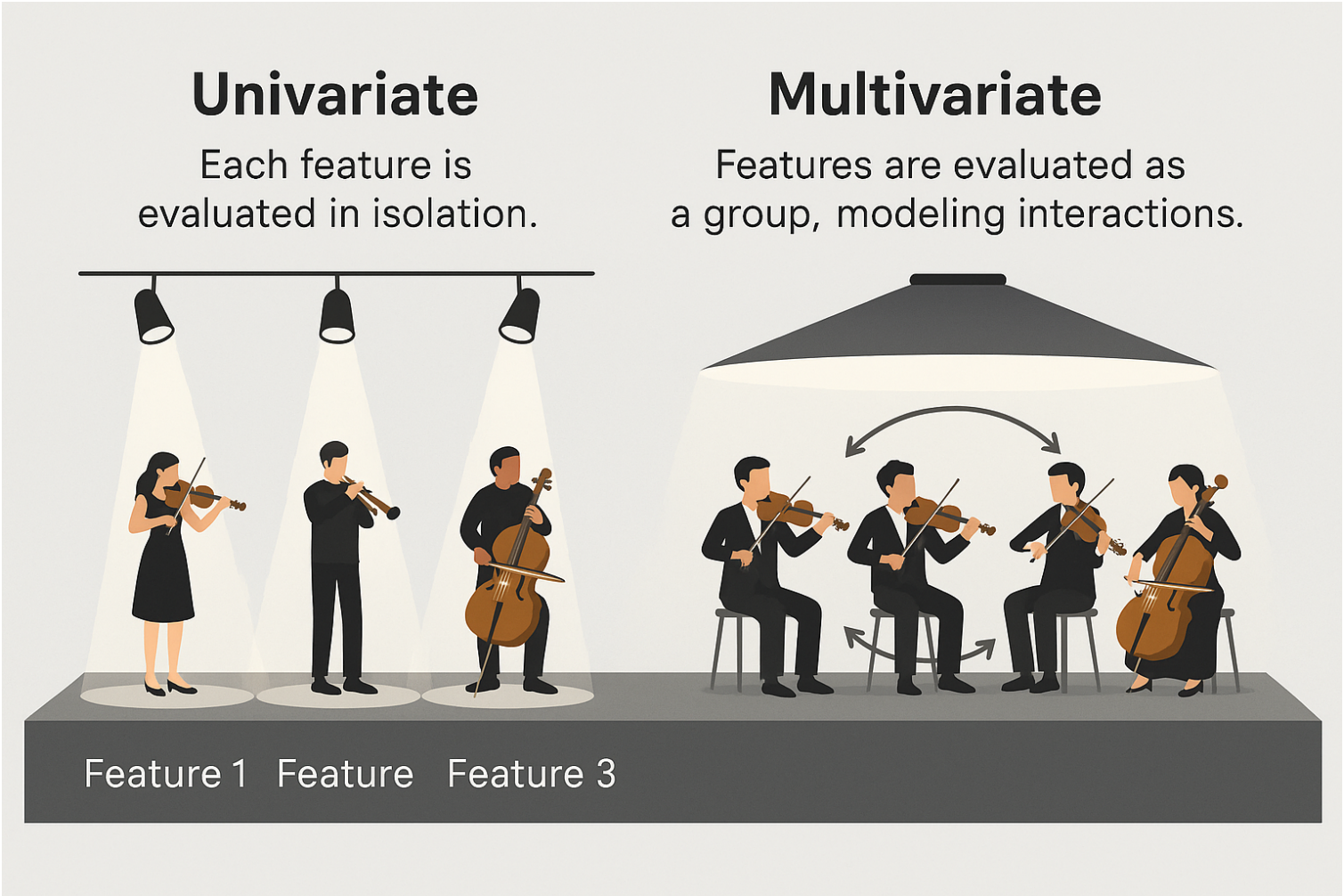
Univariate:

- Evaluate each feature independently.
- Efficient but ignores feature interactions.

Multivariate:

- Evaluate feature subsets jointly.
- Can model dependencies between features.

Univariate vs Multivariate

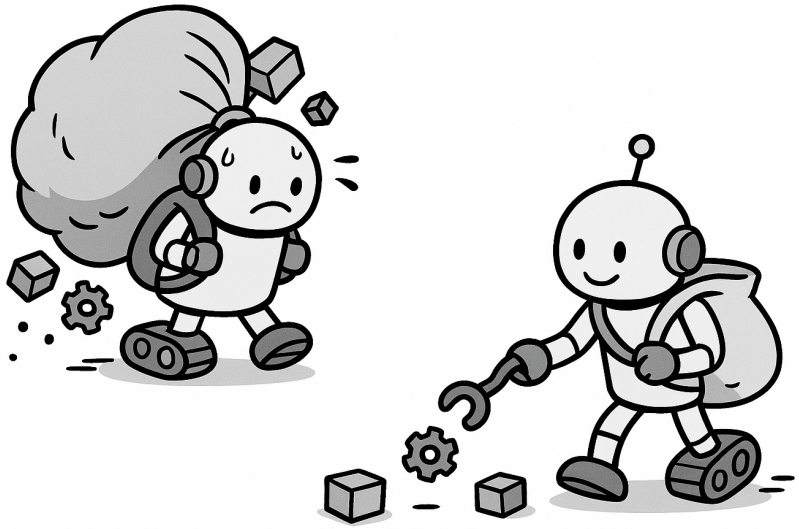


Search Direction in FS Algorithms

- **Forward Selection:** Start empty, add features incrementally.
- **Backward Elimination:** Start full, remove features step-by-step.
- **Bidirectional Search:** Combine forward and backward moves.

Heuristics:

- Greedy, Genetic Algorithms, Simulated Annealing, Beam Search



Algorithms

Feature Selection Algorithms

We now review some important algorithms used in practice.

- Filters: Kruskal-Wallis, Mutual Information, mRMR, ReliefF
- Wrappers: SVM-RFE, SVM + Genetic Algorithm
- Embedded: LASSO, Decision Tree, Random Forest

Kruskal-Wallis Filter

Non-parametric filter based on ranked data.

- Tests if distributions differ across classes.
- Used to rank features by relevance.

Statistic:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)$$

Mutual Information Filter

Measures dependency between feature X and label Y.

$$I(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

- Captures non-linear relationships.
- Higher MI = more relevant feature.

Minimum Redundancy Maximum Relevance (mRMR)

Select features that are:

- Highly relevant to the class (high MI with label)
- Minimally redundant (low MI with selected features)

$$\max_{x_j \in X - S} \left[I\left(x_j; c\right) - R(x_j, S) \right] \tag{1}$$

Where X is the set of all attributes, S is the set of already selected features, I is the mutual information function, and R is the redundancy function.

ReliefF

Estimates feature relevance by comparing near-hit and near-miss samples.

- Handles multiclass, noisy, and incomplete data.
- Score based on feature value differences between neighbors.

Weight update:

$$W[A] \leftarrow W[A] - \mathit{diff}(A, R, H) + \mathit{diff}(A, R, M)$$

ReliefF

1. set all weights $W[A] := 0.0$;
2. **for** i := 1 **to** m **do begin**
3. randomly select an instance R_i;
4. find k nearest hits H_j;
5. **for** each class $C \neq class(R_i)$ **do**
6. from class C find k nearest misses $M_j(C)$;
7. **for** $A := 1$ **to** a **do**
8. $W[A] := W[A] - \sum_{j=1}^{k} \text{diff}(A, R_i, H_j)/(m \cdot k) +$
9. $\sum_{C \neq class(R_i)} [\frac{P(C)}{1-P(class(R_i))} \sum_{j=1}^{k} \text{diff}(A, R_i, M_j(C))]/(m \cdot k)$;
10. **end**;

SVM-RFE (Recursive Feature Elimination)

Wrapper method using weights from linear SVMs.

- Iteratively removes least relevant feature.
- Ranks features by their effect on SVM margin.

Score:

$$score(f_i) = w_i^2$$

RFE

Algorithm 2 RFE Pseudo Code

Input : **S**: The set of all attributes
Output: **E**: reverse ordered ranked list of features

$E \leftarrow emptyList$
for $i \leftarrow 1$ **to** $|S|$ **do**
    $w = select\_worst\_attribute(S)$
    $S \leftarrow S$ *remove* w
    $E \leftarrow E$ *append* w
end
return E

SVM + Genetic Algorithm

Hybrid approach using SVM accuracy as fitness.

- Chromosomes encode feature subsets.
- GA operations evolve toward optimal subset.
- Evaluated using SVM performance.

LASSO (Least Absolute Shrinkage and Selection Operator)

Linear model with ℓ_1 regularization to induce sparsity.

$$\min_{\beta_0,\beta} \left\{ \frac{1}{N} \|y - \beta_0 - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{2}$$

Where y is the outcome, X is the independent variables, N is the number of instances in the data, β is the unknown parameters to be calculated, and λ is the regularization term.

- Shrinks some weights to 0, effectively selecting features.
- Efficient and interpretable.

Decision Tree

Tree-based model that splits on important features.

- Feature importance derived from information gain.
- Naturally selects discriminative features.

Gini Importance:

$$Importance(f) = \sum_{nodes} \Delta Gini(f)$$

Random Forest

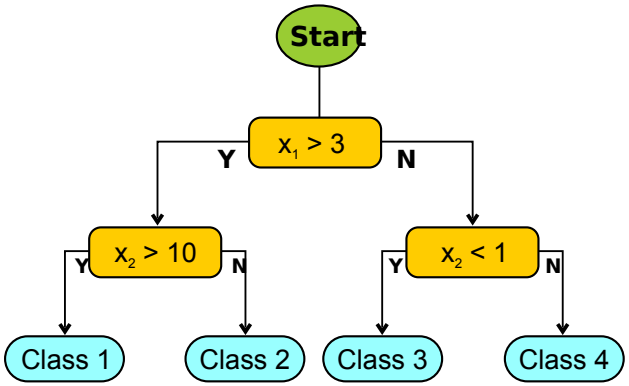
Ensemble of decision trees trained on random subsets.

- Importance = average decrease in impurity across trees.
- Reduces overfitting and improves generalization.

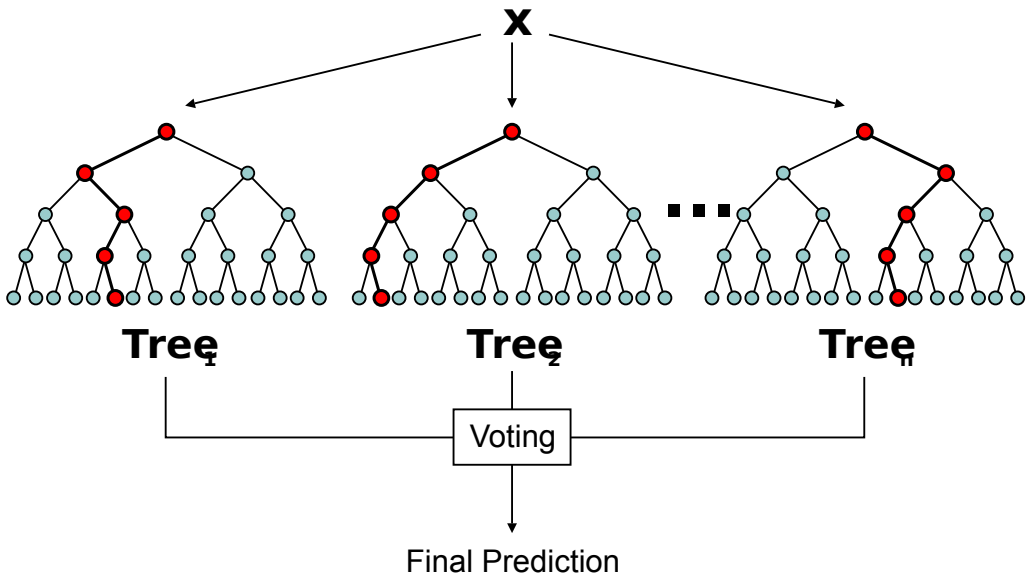
Feature ranking:

$$Importance(f) = \frac{1}{T} \sum_{t=1}^{T} Importance_t(f)$$

Trees



(a) Decision Tree



(b) Random Forest

Feature selectors properties

Algorithm	Algorithm Type	Selection Format			Attributes Correlation		Search Type
		Subset	Rank	Weights	Univariate	Multivariate	
KW Filter	Filter			✓	✓		-
MI Filter	Filter			✓	✓		-
mRMR	Filter			✓		✓	Additive
ReliefF	Filter			✓		✓	-
SVM-RFE	Wrapper		✓			✓	Subtractive
SVM-GA	Wrapper	✓				✓	Both
Decision Tree	Embedded			✓		✓	-
Lasso	Embedded			✓		✓	-
Linear SVM	Embedded			✓		✓	-
ReliefF-GA	Hybrid	✓				✓	Both
Random Forest	Ensemble			✓		✓	-