DECISION TREES AND RANDOM FORESTS: FUNDAMENTALS AND ENSEMBLE METHODS

AGENDA

1. **Decision Trees**
    - Fundamentals
    - Structure & Prediction
    - Building Process
    - Examples & Graphs
    - Splitting Criteria
    - Pruning & Parameters
2. **Random Forests**
    - Fundamentals
    - Structure & Prediction
    - Building Process
    - Key Components
    - Parameters & Advantages
3. **Key Takeaway**
    - When to Use
    - Performance Considerations
    - Advanced Techniques

WHAT ARE DECISION TREES?

Supervised models for classification and regression.

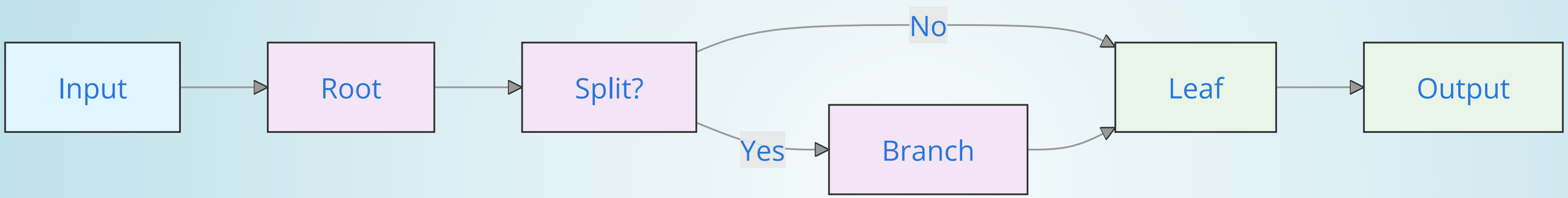
- Tree-like structure: Root to leaves
- Non-parametric: No data assumptions
- Interpretable: Easy to visualize decisions

DECISION TREE STRUCTURE AND PREDICTION

- **Root Node**: Full dataset, first split
- **Internal Nodes**: Feature thresholds (e.g., Age > 30?)
- **Leaf Nodes**: Predictions (class or value)

Depth shows decision complexity.

Prediction Flow



Traverse from root to leaf.

HOW DECISION TREES WORK

Recursive splitting of data:

1. Select best feature at root
2. Split using criteria (Gini/entropy for class, MSE for regression)
3. Recurse until stop (e.g., max depth)
4. Predict: Majority class or mean value
5. Prune to avoid overfitting

TREE EXAMPLES

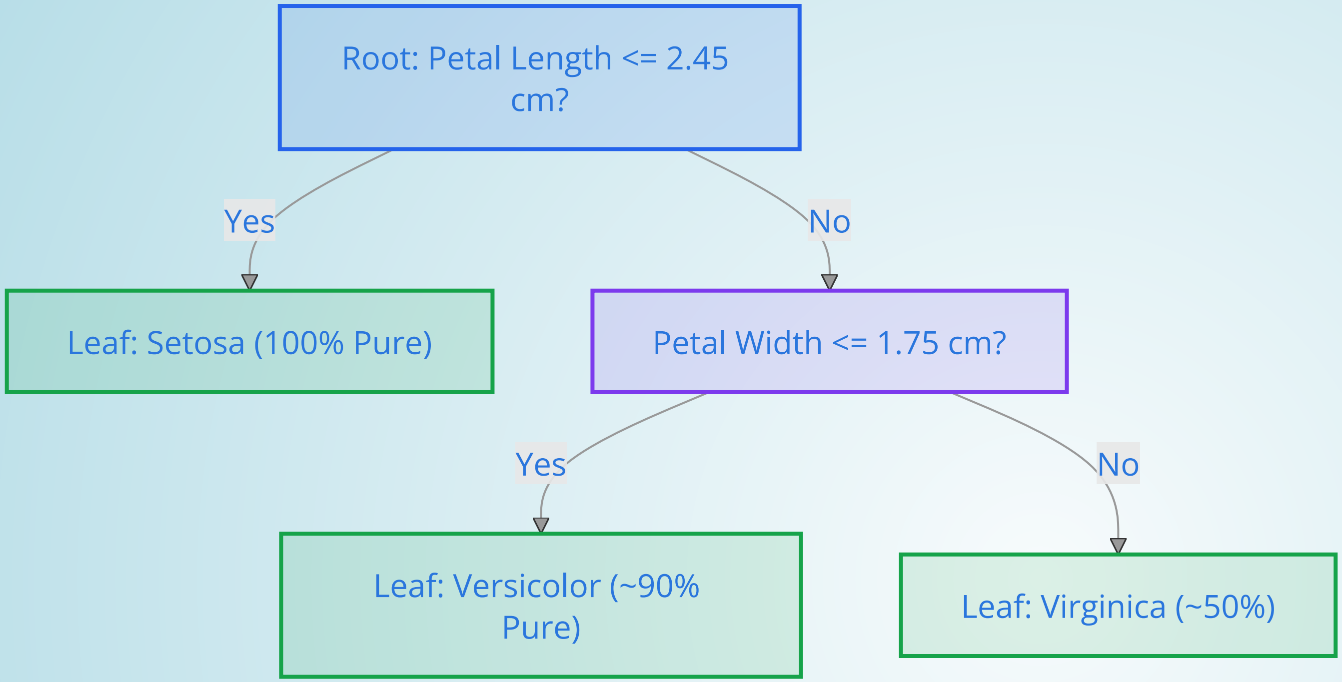
Classification (Iris):

- Root: Petal length > 2.5 cm?
- Yes: Petal width split → Versicolor
- No: Setosa

Regression (House Prices):

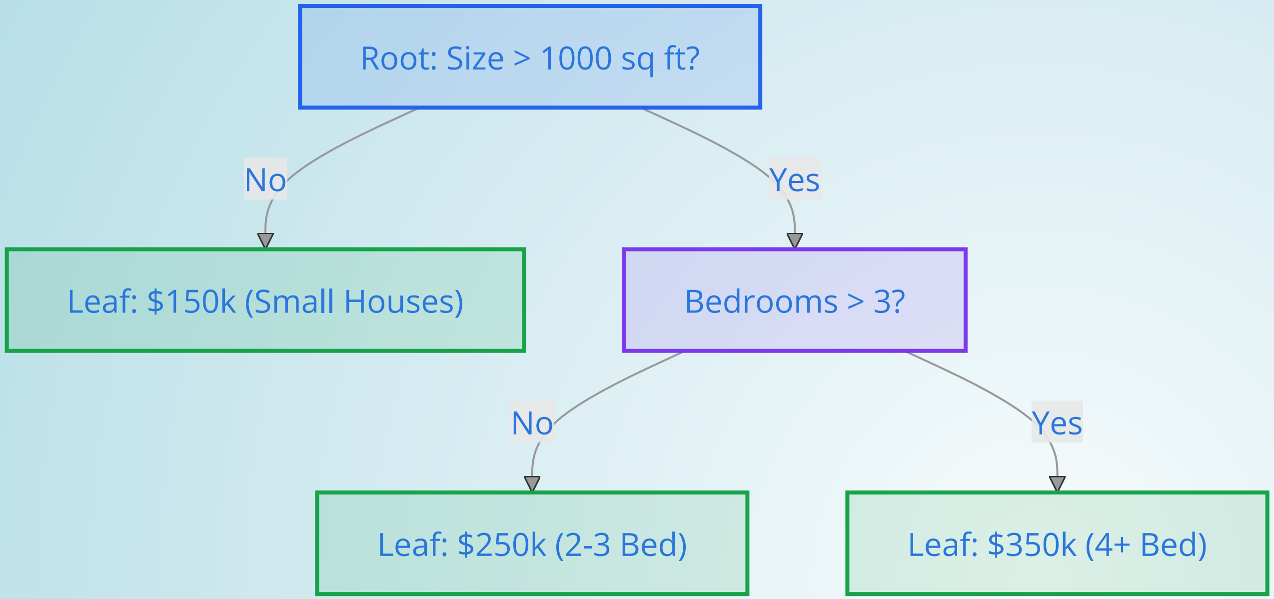
- Root: Size > 1000 sq ft?
- Yes: $300k mean
- No: $150k mean

DECISION TREE GRAPH (IRIS)



Partitions data by features to pure leaves.

REGRESSION TREE GRAPH (HOUSE PRICES)



Leaves hold mean values for predictions.

SPLITTING CRITERIA

Choose splits to maximize purity/minimize error.

Classification:

- Gini Impurity: Measures misclassification risk (0 pure, 0.5 max impure)
- Entropy: Measures uncertainty; aim for max information gain
- Gini faster; both similar results

Regression:

- MSE: Average squared error; sensitive to outliers
- MAE: Average absolute error; robust to outliers

Default: Gini for class, MSE for regression.

PRUNING AND REGULARIZATION

Prevents overfitting:

- **Pre-pruning**: Stop early (max depth, min samples)
- **Post-pruning**: Trim after building

Type	Pros	Cons
Pre	Fast	May underfit
Post	Accurate	Slower

Use min_samples_leaf for smoothing.

DT PARAMETERS

Parameter	Description	Impact
Max Depth	Tree levels	Deeper = more fit, risk overfit
Min Samples Split	For internal nodes	Higher = less overfit
Min Samples Leaf	For leaves	Smooths predictions

Tune with grid search/CV.

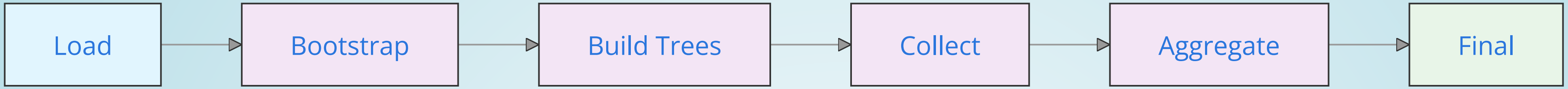
WHAT ARE RANDOM FORESTS?

Ensemble of decision trees (extension of DT).

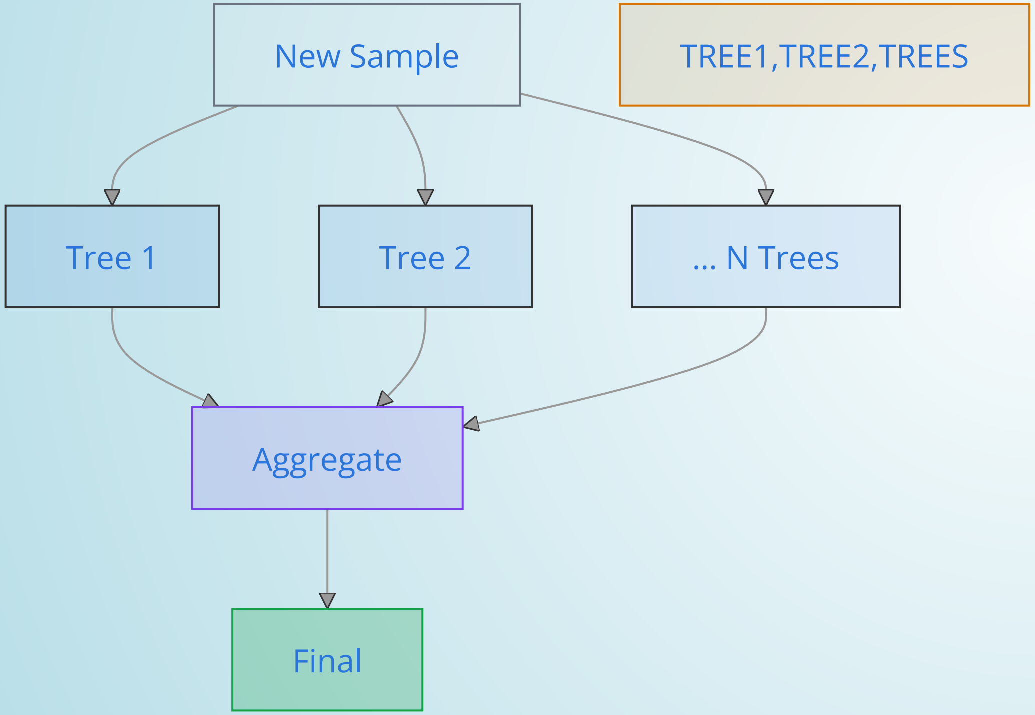
- Combines multiple trees for better accuracy
- Reduces overfitting via diversity (bagging + random features)
- Voting/averaging for predictions
- Feature importances for insights
- OOB error for built-in validation

RF STRUCTURE AND PREDICTION

Structure Graph (Simplified):



Prediction Flow:



Multiple trees to consensus; parallel predictions.

HOW RANDOM FORESTS WORK

1. Bootstrap samples (bagging: ~63% unique data per tree)
2. Random features per split (sqrt(n) for class, n/3 for reg)
3. Build independent trees (using DT criteria)
4. Aggregate: Majority vote (class) or average (reg)
5. Less greedy splits reduce variance

OOB: Unused data (~37%) for quick validation.

RF KEY COMPONENTS AND PRUNING

Components:

- Bagging: Reduces variance
- Random Features: Ensures diversity
- Aggregation: Combines outputs
- Feature Importance: Avg impurity decrease
- OOB Error: Internal validation

RF PARAMETERS AND ADVANTAGES

Parameters:

Parameter	Description	Impact
N Estimators	# Trees	More = stable, slower
Max Features	Per split	sqrt(n) class; /3 reg
Max Depth	Per tree	Controls complexity
Bootstrap	Sampling	True for diversity

Tune: Grid search, monitor OOB.

Advantages over DT:

- Less overfitting (averaging)
- Higher accuracy on tabular/noisy data
- Feature rankings
- Handles outliers better
- Parallelizable

WHEN TO USE

- **Decision Trees**: Interpretable models, small/medium data, explainability key (e.g., medical decisions)
- **Random Forests**: Noisy/high-dimensional data, accuracy priority (e.g., finance, customer analytics)
- **Both**: Non-linear/tabular problems; avoid for sequential data (use RNNs)
- Imbalanced: RF with weights/sampling
- Quick Insights: Trees for rules; RF for rankings

PERFORMANCE CONSIDERATIONS

- **Training**: Trees O(n log n); RF O(n_estimators * n log n), parallelizable
- **Prediction**: Trees O(depth); RF O(n_estimators * depth), faster with fewer trees
- **Memory**: Scales with trees; store essentials
- **Scalability**: RF handles 1000s features; subsample for millions samples
- **Bias-Variance**: Trees high variance; RF low via averaging

ADVANCED TECHNIQUES

- **Gradient Boosting**: Sequential trees (XGBoost, LightGBM) for higher accuracy
- **Extra Trees**: RF variant with random splits for speed
- **Feature Selection**: Use RF importances iteratively
- **Hybrids**: Stack RF with NNs or pipelines
- **Extensions**: Isolation Forests for anomalies; RF for time series

CONCLUSION

- **Decision Trees**: Foundational, interpretable for understanding decisions
- **Random Forests**: Robust ensembles for accurate, production-ready predictions
- **Key Takeaway**: Start simple with trees, scale to RF; always tune and validate

Apply these for versatile ML on tabular data!