Random Forest - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

Random Forests are ensembles of decision trees trained on randomized views of the data and feature space. Two sources of randomness—(i) *bootstrapping* the training set for each tree and (ii) *feature subsampling* at each split—decorrelate the base learners. Averaging (regression) or majority vote (classification) then reduces variance and improves robustness. We derive the node-splitting criteria (impurity decreases), the ensemble construction via bagging and random subspaces, aggregation rules, and variance/correlation effects, and we specify all variables with dimensions.

1.2 Data and Notation

Let the training set be

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \quad \mathbf{x}_i \in \mathbb{R}^d \text{ (column)}, \qquad y_i \in \begin{cases} \{1, 2, \ldots, K\} & \text{(classification)}, \\ \mathbb{R} & \text{(regression)}. \end{cases}$$

A Random Forest contains $T \in \mathbb{N}$ trees. At any node t of a tree, n_t denotes the number of samples reaching t, and $p_{k,t}$ the empirical class proportion (classification).

Dimensions: $n, d, K, T \in \mathbb{N}$ (scalars); $\mathbf{x}_i \in \mathbb{R}^d$ (dimension $d \times 1$); y_i scalar.

1.3 Model Formulation: Decision Trees as Base Learners

A decision tree recursively partitions $\mathbb{R}^d$ into disjoint leaf regions $\{R_\ell\}_{\ell=1}^{L}$ and predicts a constant per leaf:

$$f_{\text{tree}}(\mathbf{x}) = \sum_{\ell=1}^{L} c_\ell \, \mathbf{1}\{\mathbf{x} \in R_\ell\}, \quad \begin{cases} c_\ell = \arg\max_k \; p_{k,\ell} & \text{(classification)}, \\ c_\ell = \frac{1}{|R_\ell|} \sum_{\mathbf{x}_i \in R_\ell} y_i & \text{(regression)}. \end{cases}$$

Impurity measures and split selection. At node t with children t_L, t_R and sizes n_L, n_R ($n_L + n_R = n_t$), define:

$$\text{Gini:} \quad I_G(t) = 1 - \sum_{k=1}^{K} p_{k,t}^2, \qquad \text{Entropy:} \quad H(t) = -\sum_{k=1}^{K} p_{k,t} \log p_{k,t},$$

$$\text{Variance (regression):} \quad V(t) = \frac{1}{n_t} \sum_{i \in t} \big(y_i - \overline{y}_t\big)^2, \;\; \overline{y}_t = \frac{1}{n_t} \sum_{i \in t} y_i.$$

If a node t is split into two child nodes t_L (left) and t_R (right) with n_L and n_R samples respectively, the impurity decrease for a candidate split is

$$\Delta \mathsf{Imp} \;=\; \mathsf{Imp}(t) \;-\; \frac{n_L}{n_t}\, \mathsf{Imp}(t_L) \;-\; \frac{n_R}{n_t}\, \mathsf{Imp}(t_R),$$

with $\mathsf{Imp} \in \{I_G, H, V\}$. Choose the split maximizing $\Delta \mathsf{Imp}$; stop by a criterion (e.g., min leaf size or max depth).

Properties: $p_{k,t} \in [0,1]$ dimensionless; I_G, H, V are scalars; $I_G = H = V = 0$ on pure/constant leaves.

1.4 Ensemble Construction: Bagging + Random Subspaces

For each tree $t = 1, \ldots, T$:

(i) Bootstrapping (bagging). Draw a bootstrap sample $\mathcal{D}^{(t)} = \{(\mathbf{x}_i^{(t)}, y_i^{(t)})\}_{i=1}^{n}$ by sampling *with replacement* from $\mathcal{D}$, where each $\mathbf{x}_i^{(t)} \in \mathbb{R}^d$ and $y_i^{(t)} \in \{1, \ldots, K\}$.

(ii) Random feature selection at each split. At each node, sample a feature subset $\mathcal{F} \subset \{1, \ldots, d\}$ with $|\mathcal{F}| = m$ (e.g., $m = \sqrt{d}$ for classification, $m = d/3$ for regression) and evaluate $\Delta \mathsf{Imp}$ only for features in $\mathcal{F}$. This reduces inter-tree correlation while keeping trees deep (low bias).

(iii) Leaf prediction. Assign c_ℓ as majority class (classification) or mean response (regression) within each leaf region.

1.5 Prediction Aggregation

Given trained trees $\{h^{(t)}(\cdot)\}_{t=1}^{T}$:

Classification. Per-tree label $h^{(t)}(\mathbf{x}) \in \{1, \ldots, K\}$; forest posterior and decision

$$\widehat{p}(Y = k \mid \mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{h^{(t)}(\mathbf{x}) = k\}, \qquad \hat{y}(\mathbf{x}) = \arg\max_k \widehat{p}(Y = k \mid \mathbf{x}).$$

Regression. Per-tree real output $h^{(t)}(\mathbf{x}) \in \mathbb{R}$; forest prediction and dispersion

$$\hat{y}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} h^{(t)}(\mathbf{x}), \qquad \widehat{\mathrm{Var}}(\mathbf{x}) = \frac{1}{T-1} \sum_{t=1}^{T} \big(h^{(t)}(\mathbf{x}) - \hat{y}(\mathbf{x})\big)^2.$$

1.6 Why Forests Generalize: Variance & Correlation

Let $f_t(\mathbf{x})$ be the prediction of tree t in regression at a fixed $\mathbf{x}$, with $\mathrm{Var}(f_t) = \sigma^2$ and pairwise correlation ρ across trees. Then the ensemble average satisfies

$$\mathrm{Var}\left[\frac{1}{T} \sum_{t=1}^{T} f_t\right] \;=\; \rho\, \sigma^2 \;+\; \frac{1-\rho}{T}\, \sigma^2.$$

Thus reducing inter-tree correlation ρ (via bootstrapping and feature subsampling) can be as important as increasing T.

Out-of-bag (OOB) facts. For each index i, $\Pr\big((\mathbf{x}_i, y_i) \notin \mathcal{D}^{(t)}\big) = (1 - \frac{1}{n})^n \approx e^{-1} \approx 0.368$. Let $\mathcal{B}_i = \{t : i \text{ is OOB for tree } t\}$. Define the OOB prediction

$$\widehat{y}_i^{\text{OOB}} = \begin{cases} \frac{1}{|\mathcal{B}_i|} \sum_{t \in \mathcal{B}_i} h^{(t)}(\mathbf{x}_i), & \text{regression}, \\ \arg\max_k \ \frac{1}{|\mathcal{B}_i|} \sum_{t \in \mathcal{B}_i} \mathbf{1}\{h^{(t)}(\mathbf{x}_i) = k\}, & \text{classification}, \end{cases}$$

and OOB error $\frac{1}{n} \sum_{i=1}^{n} \ell(\widehat{y}_i^{\text{OOB}}, y_i)$, which estimates generalization error without a held-out set.

1.7 Feature Importance

Impurity-based (in-bag). Sum impurity reductions attributable to feature j across all splits where j is used, weighting by node sample fraction.

Permutation (OOB-based). For each tree, permute feature j among its OOB samples, recompute OOB error, and take the increase as $\text{Imp}_j^{\text{perm}}$. Larger increases imply greater predictive contribution.

1.8 Algorithm (Random Forest: TDIDT + Randomization)

1. **Input:** $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; number of trees T; features per split m; stopping criteria (max depth, min samples per leaf).
2. **For** $t = 1$ to T:
   (a) Draw bootstrap sample $\mathcal{D}^{(t)}$ of size n (with replacement).
   (b) Grow a tree on $\mathcal{D}^{(t)}$:
      i. At each node, sample $\mathcal{F} \subset \{1, \ldots, d\}$ with $|\mathcal{F}| = m$.
      ii. For candidates using features in $\mathcal{F}$, compute $\Delta\mathsf{Imp}$ and split on the maximizer.
      iii. Recurse until stopping; set leaf c_ℓ as majority class (classification) or leaf mean (regression).
3. **Predict:** Aggregate per-tree predictions by majority vote (classification) or averaging (regression).
4. **(Optional) OOB:** Compute OOB predictions/errors and permutation importances.

1.9 Summary of Variables and Their Dimensions

- $\mathbf{x}_i \in \mathbb{R}^d$: ith input (dimension $d \times 1$); $y_i \in \{1, \ldots, K\}$ or $\mathbb{R}$.
- n: number of training samples; d: number of features; K: number of classes; T: number of trees.
- At node t: n_t samples; class proportions $p_{k,t} \in [0, 1]$.
- Impurities: $I_G(t)$ (Gini), $H(t)$ (entropy), $V(t)$ (variance); all scalars.
- Feature subset per split: $\mathcal{F} \subset \{1, \ldots, d\}$ with $|\mathcal{F}| = m$.
- Tree leaves: regions $R_\ell \subset \mathbb{R}^d$, predictions $c_\ell \in \{1, \ldots, K\}$ or $\mathbb{R}$.
- Ensemble predictions: $h^{(t)}(\mathbf{x})$ per tree; forest $\hat{y}(\mathbf{x})$ by vote/average.

1.10 Summary

From first principles and impurity-based derivations: (i) grow deep decision trees by maximizing impurity reduction (Gini/entropy for classification, variance for regression); (ii) bag the training set and (iii) restrict candidate features at each split to decorrelate trees; (iv) aggregate by voting/averaging. Variance analysis shows that averaging reduces noise while feature subsampling lowers inter-tree correlation, jointly driving strong generalization. All variables above are defined with their dimensions for clarity and implementation.