# Exploring Bagging and Boosting on the Breast Cancer Wisconsin Dataset

Student Name: Dinesh Singh Pheiroijam
Student ID: 23085168
GitHub:  Exploring_Bagging_and_Boosting_on_the_Breast_Cancer_Wisconsin_Dataset
Dataset: Breast Cancer Wisconsin (Diagnostic)

## 1. Introduction

Ensemble learning combines multiple “weak” models to build a stronger predictor, often improving robustness and accuracy compared with a single model. This tutorial focuses on two popular ensemble strategies—**bagging** and **boosting**—applied to the Breast Cancer Wisconsin (Diagnostic) dataset, a binary classification problem of predicting whether a breast tumor is malignant or benign.

The goal is to answer a practical question: ***Does bagging or boosting give better generalization than a single decision tree on this medical dataset, and how does ensemble size affect performance and overfitting?*** The tutorial walks through data exploration, a baseline decision tree, Random Forest (bagging), Gradient Boosting (boosting), and an analysis of errors, feature importance, and ethical considerations.

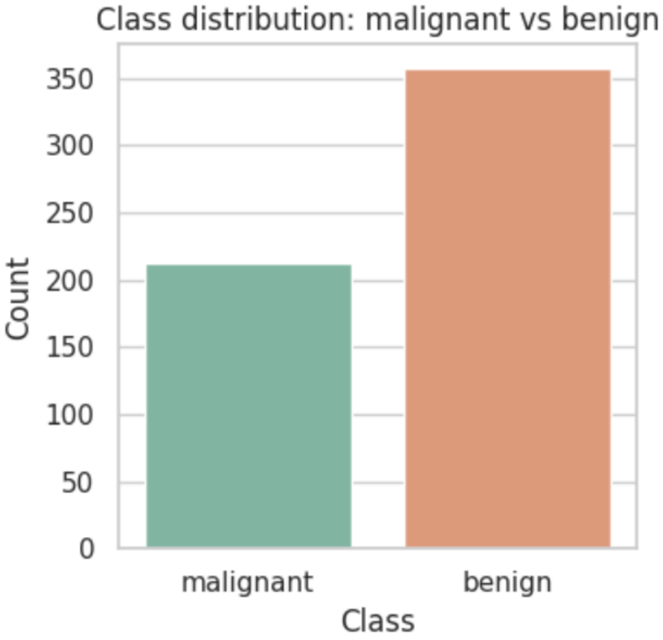
## 2. Dataset and Problem Setup

## 2.1 Dataset description

The Breast Cancer Wisconsin (Diagnostic) dataset contains **569** patient records, each with **30 continuous features** that summarize properties of cell nuclei computed from digitized images of fine needle aspirates of breast masses. The target has two classes: **malignant (0)** and **benign (1)**, which are mapped to human-readable labels for interpretability in this tutorial.

We combined the feature matrix into a pandas DataFrame and added two target columns: a numeric `target` and a string `target_label` indicating “malignant” or “benign”.

**Figure 1 – Class distribution bar chart**



## 2.2 Class balance and data split

The dataset contains **357 benign** and **212 malignant** samples, so benign tumors represent about **62.74%** of the data and malignant tumors about **37.26%**, giving a moderately imbalanced distribution where accuracy alone can be misleading.

To support fair model selection, the data was split into **60% training (341 samples)**, **20% validation (114 samples)**, and **20% test (114 samples)** using stratified splits to preserve the malignant/benign ratio in each subset. The training set is used to fit models, the validation set to tune ensemble hyperparameters and compare bagging versus boosting, and the test set is held out for final evaluation.

---

# 3. Methods: Single Tree, Bagging, and Boosting

## 3.1 Baseline: decision tree classifier

As a baseline, a `DecisionTreeClassifier` with default settings is trained on the training set and evaluated on validation and test sets. The tree reaches **1.000** training accuracy, **0.956** validation accuracy, and **0.904** test accuracy, indicating that it fits the training data perfectly but loses performance on unseen data due to overfitting.

This behavior illustrates a key motivation for ensembles: a single deep tree has high variance—small changes in the training data can lead to very different trees and test performance—so it is a natural candidate for bagging and boosting.

## 3.2 Bagging with Random Forest

Bagging (bootstrap aggregating) builds many base learners on different bootstrap samples and averages their predictions to reduce variance. A **Random Forest** implements bagging with decision trees by sampling both rows (bootstrap samples) and features at each split, encouraging diversity among trees.

In this tutorial, a `RandomForestClassifier` with **200 trees**, unlimited depth, and a fixed random seed achieves **1.000** training accuracy, **0.956** validation accuracy, and **0.947** test accuracy, outperforming the single tree on the held-out test set. The improvement in test accuracy without changing validation accuracy suggests that averaging many decorrelated trees successfully reduces variance while preserving predictive power.

## 3.3 Boosting with Gradient Boosting

Boosting builds an ensemble sequentially, where each new tree focuses on correcting the residual errors of the previous ensemble, reducing bias at the cost of increased sensitivity to noise and hyperparameters. A `GradientBoostingClassifier` with **200 shallow trees** (depth 3) and learning rate 0.1 reaches **1.000** training accuracy, **0.947** validation accuracy, and **0.930** test accuracy on this dataset.

While Gradient Boosting improves on the single tree baseline, it performs slightly worse than the Random Forest on the test set with the chosen hyperparameters, highlighting the importance of tuning ensemble size and learning rate for boosting methods.

---

# 4. Results and Comparative Analysis

## 4.1 Accuracy comparison

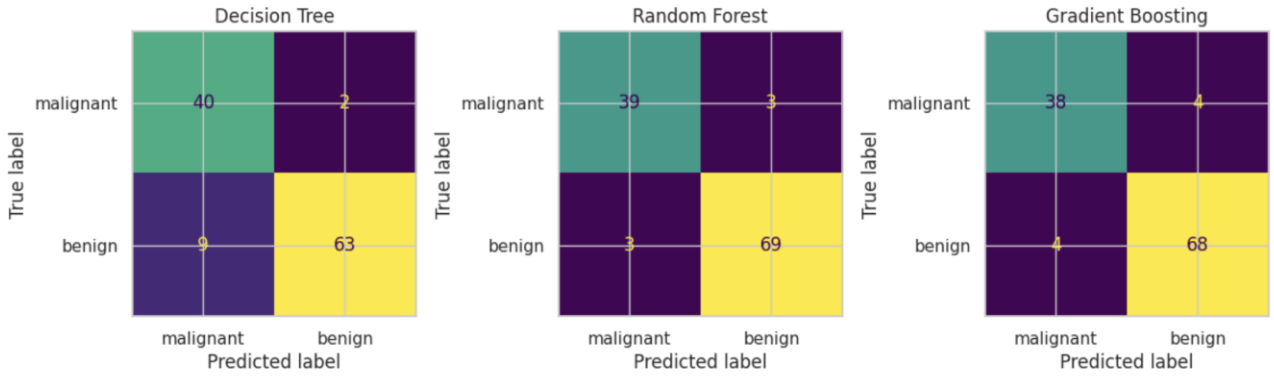
**Table 1 – Accuracy of baseline and ensemble models**

| Model | Train accuracy | Validation accuracy | Test accuracy |
|---|---|---|---|
| Decision Tree | 1.000 | 0.956 | 0.904 |
| Random Forest (bagging) | 1.000 | 0.956 | 0.947 |
| Gradient Boosting (boosting) | 1.000 | 0.947 | 0.930 |

All models achieve perfect training accuracy, but the single tree shows the largest drop between training and test, consistent with high variance and overfitting. Both ensemble methods generalize better, with Random Forest delivering the highest test accuracy, suggesting that bagging is particularly effective for this dataset while the chosen boosting configuration slightly overfits.

## 4.2 Confusion matrices and error types

**Figure 2 – Confusion matrices for Decision Tree, Random Forest, and Gradient Boosting**



On the test set, the Random Forest correctly classifies **39 malignant** and **69 benign** tumors, with **3 malignant tumors misclassified as benign** and **3 benign tumors misclassified as malignant**. The corresponding classification report reports precision and recall of about **0.93** for malignant and **0.96** for benign, with overall accuracy **0.95**, indicating strong but not perfect performance.

From a clinical perspective, false negatives (malignant predicted as benign) are more dangerous than false positives, because they may delay treatment. This suggests that, in real deployment, decision thresholds or class weights might be adjusted to prioritize recall for malignant cases even at the cost of more benign false positives.

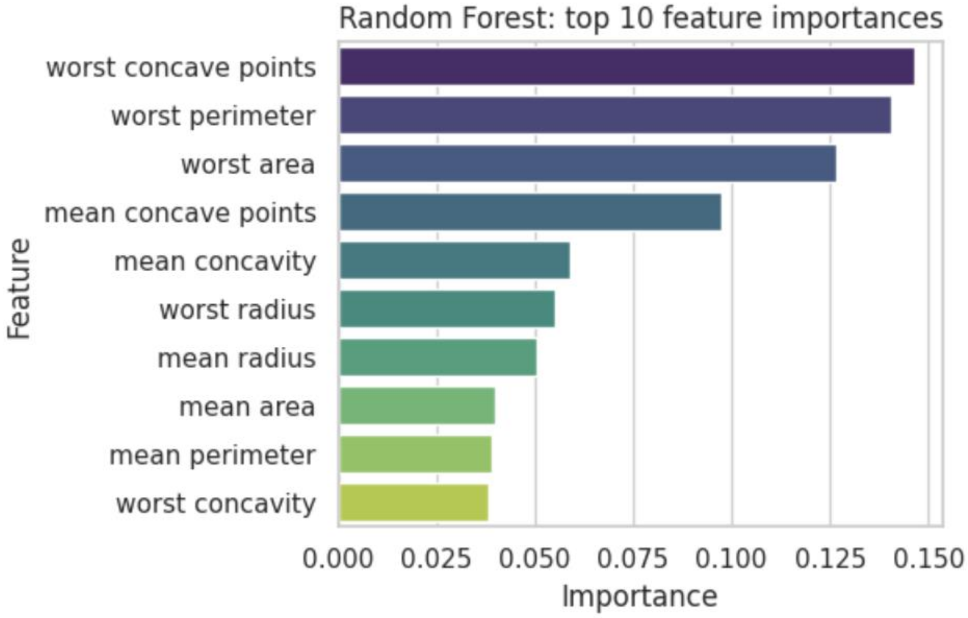
---

# 5. Understanding Ensemble Behavior

## 5.1 Feature importance in Random Forest

Random Forest feature importance indicates that **worst concave points**, **worst perimeter**, and **worst area** contribute most strongly to predictions, followed by several mean concavity and radius measures. These features capture the size and irregularity of the tumor, which is consistent with medical intuition that larger and more irregular masses are more likely to be malignant, although domain experts should validate whether the learned importance aligns with clinical evidence.

**Figure 3 – Top 10 Random Forest feature importances**



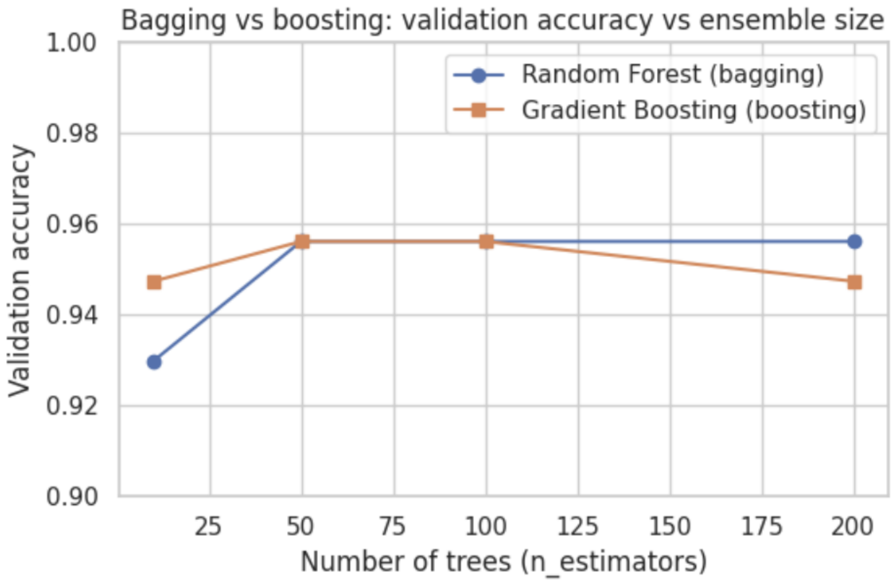
This analysis also illustrates a strength of tree-based ensembles: they provide straightforward feature importance scores, offering some interpretability compared with “black-box” models such as many deep neural networks.

## 5.2 Effect of ensemble size: bagging vs boosting

To study how ensemble size affects performance, both Random Forest and Gradient Boosting were trained with **10, 50, 100, and 200 trees**, and validation accuracy was recorded. Random Forest validation accuracy increased from **≈0.93** at 10 trees to **≈0.956** at 50 trees and then remained stable, whereas Gradient Boosting peaked around **0.956** at 50–100 trees and dropped slightly to **≈0.947** at 200 trees.

The curves confirm that bagging primarily reduces variance and quickly reaches a plateau as more trees are added, making Random Forest relatively robust to the exact number of estimators on this dataset. In contrast, boosting is more sensitive to ensemble size and can begin to overfit when too many learners are added without reducing the learning rate or applying stronger regularization.

**Figure 4 – Validation accuracy vs number of trees**



# 6. Ethical and Accessibility Considerations

## 6.1 Ethical aspects of medical prediction

Using machine learning models for cancer diagnosis raises ethical questions about patient safety, fairness, and transparency. Even though this tutorial uses a public dataset in a research context, the presence of false negatives in the Random Forest predictions illustrates that automated systems should not be used in isolation to make treatment decisions and must be validated rigorously on diverse, representative populations.

Dataset documentation should clarify how the data were collected, which patient groups are represented, and which are missing, as under-represented subgroups may experience systematically worse performance, leading to inequitable outcomes if the model is deployed.

## 6.2 Accessibility of the tutorial and code

To make the tutorial accessible, all plots use high-contrast, colour-blind-friendly palettes, and each figure is accompanied by descriptive alt-text so that screen-reader users can understand the visual content. The Jupyter notebook is structured with clear section headings, and all code required to reproduce the figures is provided in a GitHub repository with a README explaining installation steps, commands to run the notebook, and an explicit open-source licence indicating permitted reuse.
For a video-based version of this tutorial, subtitles and a downloadable transcript should be provided so that learners with hearing impairments can follow the content, and on-screen code should use sufficiently large fonts and clear contrast.

## 7. Practical Takeaways: When to Use Bagging vs Boosting

The experiments suggest the following simple guidelines for similar tabular medical datasets:

- If a single decision tree overfits and test accuracy is unstable, **Random Forest (bagging)** is often a strong default choice, as it reduces variance and is relatively easy to tune (number of trees, maximum depth, and feature subsampling).
- **Gradient Boosting** can achieve competitive performance, especially with careful tuning of learning rate, number of trees, and tree depth, but it is more sensitive to hyperparameters and can overfit when the ensemble becomes too large.
- In safety-critical applications, focus not only on accuracy but also on recall for the critical class (here, malignant) and on clear communication of model limitations and uncertainty.

## 8. Conclusion

This tutorial compared a single decision tree, Random Forest (bagging), and Gradient Boosting (boosting) on the Breast Cancer Wisconsin dataset, showing that both ensemble methods substantially improved test performance over the baseline tree, with Random Forest achieving the best generalization under the chosen settings. Analysis of feature importance, confusion matrices, and ensemble size demonstrated how bagging reduces variance through averaging many trees, while boosting reduces bias but may overfit if not carefully regularized.

Beyond raw performance, the tutorial highlighted ethical and accessibility considerations, emphasising that medical prediction models must be evaluated for their error patterns, fairness across populations, and usability for diverse learners. The accompanying notebook and GitHub repository provide a complete, reproducible workflow that other students can adapt to explore ensemble methods on their own data.

## 9. References

- Scikit-learn developers. “Breast Cancer Wisconsin (Diagnostic) dataset documentation.”educative+1
- Scikit-learn developers. “Ensemble methods: Random forests, bagging, and gradient boosting.”scikit-learn+1
- Articles and tutorials explaining bagging vs boosting and ensemble learning concepts.baeldung+4
- Resources on ethical AI and responsible use of machine learning in sensitive domains.turing+2
- Accessibility best-practice guidelines for educational content.linkedin+1