Assignment 4: Random Forest Analysis of Tree Size

Grant Booysen 25849646
Email: 25849646@sun.ac.za

Abstract—**The report investigated how random forest performance varies with decision-tree depth, the number of randomly selected features per split, and ensemble size across three classification datasets of increasing complexity. The goal was to track performance from underfitting with very shallow trees and few features to overfitting with deep trees and many features, identify points that maximize generalization, and characterize interactions among these hyperparameters. Experiments were conducted on (i) maximum depth from minimal to deep trees, (ii) the number of features to split on with depth fixed at the best value from (i), and (iii) the balance between the number of trees and depth. A final experiment created ensembles that mixed underfitted and overfitted trees. Models were evaluated with train/test accuracy and five-fold cross-validation. Findings were consistent: increased depth improved test accuracy up to a point, after which performance stagnated, and the optimal depth increased with problem complexity. With depth fixed, increasing the number of features per split slightly improved performance up to a plateau where medium complexity favored moderate feature subsets, while complex data benefited from larger subsets. The inclusion of more trees in the ensemble improved stability and accuracy up to a plateau, with gains moderated by tree depth. Bagging with only overfitted trees yielded either the best stability, cross-validated accuracy, test accuracy, or a combination of these across all datasets.**

Overall, the best configurations balanced depth and ensemble size, with larger values of both generally preferred as task complexity increased. The number of random features considered at each split did not significantly affect performance once depth was near optimal.

Index Terms—**Random Forest, Ensemble learning, Bagging, Decision Trees**

I. Introduction

Random forests are widely used for classification due to strong generalization, robustness, and modest tuning requirements. The performance of the model is controlled by key hyperparameters that trade off bias and variance, such as maximum decision-tree depth, the number of randomly selected features per split, and ensemble size. The report aims to provide an understanding of how these controls move the model to better performance. The idea is to identify configurations that generalize across problem complexities.

The first goal of the study is to characterize how tree depth and feature subsampling influence random forest performance, and to identify configurations that maximize test accuracy and cross-validated performance across datasets of increasing difficulty. The analysis is achieved by exploring: (i) maximum depth from minimal to deep trees at fixed ensemble size, number of random features per split, and bagging; (ii) fixing depth at the best value from (i) and exploring the optimal number of features to split on per node; (iii) depth and ensemble-size interactions; and (iv) composing ensembles that intentionally mix underfitted and overfitted trees. Evaluation uses train/test accuracy and five-fold cross-validation to better assess generalization and stability.

The report aims to provide a reproducible behavior for the bias-variance trade-off for random forests, and to extract generalizations for hyperparameter selection that adapt to dataset complexity. The design follows controlled experiments to reveal how each hyperparameter influences performance, and how the parameters interact. The analysis spans three synthetic datasets of increasing complexity to assess how optimal configurations shift with task difficulty.

The main observations are consistent with the bias-variance expectations. Increasing the depth of the trees in the ensemble improves test accuracy up to a point, after which gains stagnate; the optimal depth increases with complexity, for example, approximately 8, 15, and 20 on the simple, medium, and complex datasets, respectively. With depth fixed, increasing the number of features to split on did not drastically improve performance. The inclusion of more trees in the ensemble improves stability and accuracy up to a plateau, where the number of trees needed to reach the plateau is dependent on the depth of the trees. For mixed-depth and only overfitted ensembles, on the simple dataset, the highest test accuracy was 0.960 with a tie between the balanced and only overfitted compositions, while only overfitted attained the best cross-validated score of 0.966; on the medium dataset, the only-overfit composition achieved both the highest test accuracy of 0.685 and the best cross-validated score of 0.691; on the complex dataset, only overfitted trees achieved the highest test accuracy of 0.610 and the most stable cross-validation, with majority overfitted exhibiting a marginally higher cross-validation (CV) mean of 0.564 versus 0.563 for only overfitted.

The remainder of the report is organized as follows: Section 2 provides background; Section 3 details the methodology and implementation; Section 4 describes the empirical procedure; Section 5 presents results and discussion; Section 6 concludes with implications and future directions.

II. Background

The background section discusses the fundamental concepts and algorithms used in the report, including decision trees, ensemble learning, and random forests.

A. Decision Trees

The decision tree is a supervised learning algorithm used for classification and regression tasks. The decision tree works by recursively splitting the data into subsets based on feature values, creating a tree-like model of decisions. Each internal node represents a condition on a feature, each branch represents the outcome of the test, and each leaf node represents a class label in classification or a continuous value in regression. Decision trees are easy to interpret with the rule-based structure, but are prone to overfitting, especially with deep trees. The overfitting is mitigated in random forests by averaging predictions across many trees, which reduces variance.

B. Ensemble Learning: Bagging

Bagging, or Bootstrap Aggregating, is an ensemble learning technique that leverages the averaging of multiple models to increase the stability and accuracy of a final cohort of learners. The algorithm works by training multiple models on different subsets of the training data, which are created by randomly sampling with replacement, known as bootstrapping. The final prediction is made by aggregating the predictions of all individual models, typically by averaging for regression or majority voting for classification. Bagging helps reduce variance and combat overfitting, making the approach particularly effective for high-variance models like decision trees.

C. Random Forests

Random forests are an ensemble learning method that combines multiple decision trees to improve predictive performance and control overfitting. The key idea is to leverage the diversity of individual trees by training them on different subsets of the data and features. Traditionally, random forests use bagging to create diverse training sets for each tree. In random forests, the expectation is that each tree should be grown to the maximum depth and overfit. The high variance of each tree is then averaged out by the ensemble, leading to a final model that generalizes well. Algorithm 1 provides high-level pseudocode for the random forest algorithm.

Algorithm 1 Random Forest high-level pseudocode

```
Require: Training data $(X, y)$, n_estimators, max_depth,
    max_features, bootstrap size
for each tree $t = 1 \ldots$ n_estimators do
    Draw a bootstrap sample from $(X, y)$
    Grow a decision tree with depth limit max_depth
    At each node, consider max_features randomly selected
        features
end for
For prediction, average class probabilities across trees
```

D. Statistical comparison of models

To compare multiple model configurations evaluated on the same data partitions, non-parametric, paired statistical tests are used:

- **Friedman test**: compares k related treatments across N blocks by ranking methods within each block and testing the null hypothesis that the median ranks are equal. The test avoids normality assumptions and is appropriate for repeated-measures cross-validation.
- **Wilcoxon signed-rank test**: conducts a paired comparison between two configurations across the same blocks. The method tests whether a reference method tends to outperform a comparator. Holm correction controls the family-wise error rate over multiple pairwise tests.

These tests operate on the cross-validated fold scores used in the experiments.

III. Implementation

The implementation section describes the datasets, data preparation, and the random forest model used in the experiments.

A. Data preparation

Three synthetic, tabular classification datasets of increasing complexity were programmatically generated using `make_classification` from `sklearn.datasets` [Pedregosa et al., 2011] to enable controlled bias-variance studies and avoid data quality issues such as missing values or categorical encoding. A single random state ensured reproducibility for generation and splitting. The datasets are summarized in Table I.

Dataset	Samples	Features	(Informative, Redundant)	Classes	Overlap	Noise
Simple	500	10	(8, 2)	2	2.0	0.0
Medium	2000	25	(18, 5)	3	1.0	0.2
Complex	6000	50	(35, 10)	4	0.7	0.3

TABLE I: Summary of synthetic datasets used in experiments. The informative and redundant features are specified to control complexity. Class overlap and label noise increase with complexity.

Train/test split: A stratified 70/30 train–test split with fixed random states preserves class proportions across splits and guarantees identical partitions across runs.

Preprocessing: No scaling or normalization is applied as decision trees are insensitive to monotonic feature transforms. No imputation is required as the synthetic data contains no missing values. No label encoding is needed as the labels are already integers.

B. Random Forest model

A custom scikit-learn compatible estimator was implemented by extending `BaseEstimator` and `ClassifierMixin`. Each base learner is a `DecisionTreeClassifier`. The ensemble exposes the hyperparameters: `n_estimators`, `max_depth`, `max_features`, `bootstrap`, `criterion`, `min_samples_*`, `max_leaf_nodes`, and `ccp_alpha`. These parameters allow control over tree complexity and randomness. Each individual tree needed to be individually controlled to allow for the mixed underfit/overfit experiments.

Training uses standard bagging: for each tree, a bootstrap sample is drawn. A distinct, reproducible random seed is generated per tree from a generator seeded by the estimator's random seed. Each tree `max_features` parameter enforces feature sub-sampling at each split, mirroring the random subspace mechanism. The final classification averages the class probabilities across trees and enforces a majority vote.

The model has the following characteristics:

- Equal-weight averaging across trees with no out-of-bag estimates or class weighting by default.
- No post-hoc pruning as overfitting is intentionally permitted to study depth effects.

C. Cross-validation procedure

Across all experiments, performance for each hyperparameter setting was estimated with K-fold stratified cross-validation (CV) on the training split only using `StratifiedKFold` from `sklearn.model_selection` [Pedregosa et al., 2011], reporting the mean and standard deviation of accuracy. Concretely, for the tree-depth, feature-selection, and depth by trees grids, stratified folds for classification were used to compute cross-validated accuracy scores, and record CV mean and CV standard deviation. The held-out test split is evaluated separately to provide a single final generalization number, but not to select hyperparameters. Where an optimal depth is required to center configurations, the depth is chosen by maximizing the CV mean from the prior depth experiment.

D. Statistical testing protocol

For each dataset, repeated stratified k-fold cross-validation with five folds and seven repeats provides fold-wise accuracies for each of the five ensemble compositions. The Friedman test serves as a global check for any differences among configurations. When significant, one-sided Wilcoxon signed-rank tests compare the expected best configuration, the only overfitted trees configuration, against each alternative, with Holm-adjusted p-values to control family-wise error. Unless stated otherwise, the significance level is set to $\alpha = 0.05$, and post-hoc rejections are determined by comparing Holm-adjusted p-values against α.

IV. Empirical Procedure

The empirical procedure section details the datasets, performance measures, control parameters, and the four experiments executed to answer the study goals. All experiments are fully reproducible given the seeds and grids below.

A. Benchmarks and Splitting

The three datasets Simple, Medium, and Complex were generated as described in the Data Preparation section. Each dataset was split into a training set at a 70% split and a held-out test set at a 30% split using a stratified split to maintain class proportions.

B. Implementation and Controls

A global seed of 42 was used. Per-tree seeds were drawn from a generator seeded by the global seed value to ensure determinism. Unless varied explicitly, defaults for the decision trees were:

- Criterion for split quality was the Gini impurity.
- Minimum samples to split a node were two.
- Minimum samples per leaf was one.
- No pruning.
- Bootstrap sampling.
- Equal weighting across trees.

Dataset-specific depth grids: Each dataset had a predefined grid of maximum depths to explore:

- Simple: $\{1, 2, 3, 4, 5, 6, 8, 10, 12\}$
- Medium: $\{1, 3, 5, 7, 10, 12, 15, 18, 20, 25, 30\}$
- Complex: $\{1, 3, 5, 7, 10, 12, 15, 20, 25, 30, 35, 40\}$

C. Cross-validation for Experiments

All experiments used five-fold stratified cross-validation on the training split. For each hyperparameter configuration, the cross-validation mean and standard deviation were reported. The held-out test split was evaluated separately to report train and test accuracy. The same reproducible controls were applied across all experiments. For each configuration, the following were recorded: train accuracy, test accuracy, and CV mean and standard deviation.

D. Experiment 1: Maximum Tree Depth

The objective of Experiment 1 was to isolate the effect of maximum tree depth on generalization and overfitting. The maximum depth was grid-searched from very shallow to very deep trees while holding other factors constant. A forest of decision trees was trained, and all features are used at each split so that each tree was as strong as possible.

E. Experiment 2: Number of Features per Split

The objective was to quantify the effect of per-split feature subsampling while holding depth near the optimal values found in Experiment 1. The experiment fixed the maximum depth to the CV optimal values, where a depth of 8, 15, and 20 was used for Simple, Medium, and Complex, respectively. The size of the ensemble was limited to 30, 60, and 90 trees for the Simple, Medium, and Complex datasets respectively, and the experiment swept the maximum number of features per split over $\{1, 2, 3, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, and skipped values that exceeded each dataset's dimensionality.

F. Experiment 3: Depth versus Number of Trees

The objective was to study the interaction between ensemble size and tree depth. In the experiment, the number of features considered per split was fixed to the square root of the total number of features to provide a moderate level of randomness. Tree depth was swept over the dataset-specific depth grids while the number of trees was varied per dataset. Results were visualized as heatmaps of test accuracy on the depth of the trees versus the size of the ensemble. A further analysis was conducted to quantify how increasing the number of trees reduces variance up to a plateau and how the benefit of additional trees depends on tree depth.

G. Experiment 4: Mixed-Depth Ensembles

The experiment evaluated ensembles that mix underfitting and overfitting trees to probe bias-variance blending. Controls were fixed where the number of features per split was set to the optimal values from Experiment 2, and ensemble sizes and per-dataset maximum tree depths used the optima from Experiment 3. Five ensemble compositions of trees were compared:

- all underfitted;
- majority underfitted with 75% underfitted, 25% overfitted;
- balanced with 50% underfitted, 50% overfitted;
- majority overfitted with 25% underfitted, 75% overfitted;
- all overfitted.

Underfitted depths were sampled from depths below the optimal, and overfitted depths from just above the optimal up to the maximum. The goal was to determine whether mixing tree capacities yields improved generalization or stability compared to homogeneous ensembles of the same size.

V. Research Results

The section reports results for all four experiments. For each configuration, the cross-validated accuracy mean and standard deviation over five folds on the training split and the held-out test accuracy are reported.

A. Experiment 1: Maximum Tree Depth

Across datasets, increased maximum depth initially raised both CV and test accuracy. Beyond dataset-specific optima, test and CV curves plateaued. The figures highlight the bias-variance trade-off and visualize the depth that maximizes generalization for each dataset.

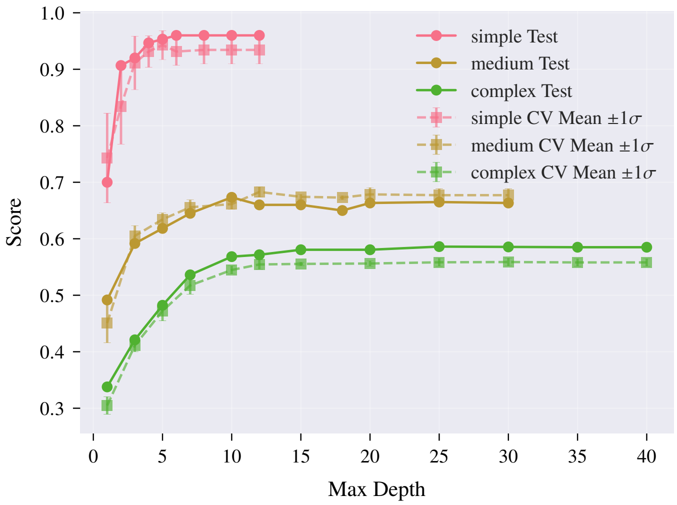
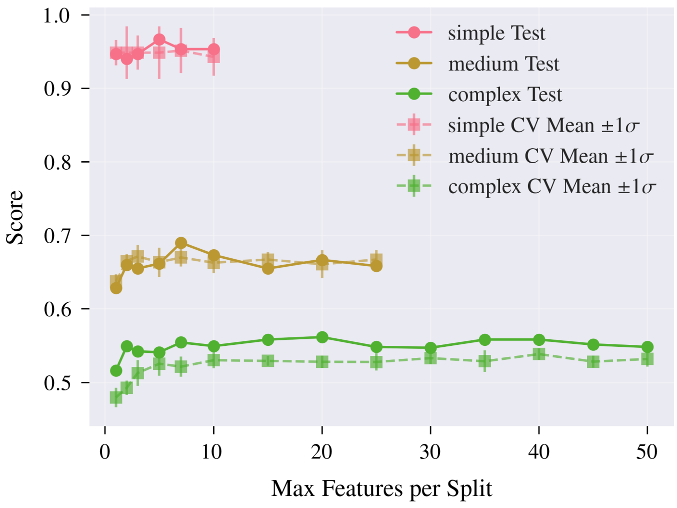


Fig. 1: Maximum tree depth versus test and cross-validated accuracy for the three datasets.

The results in Figures 1 show the relationship between maximum tree depth and both test and cross-validated accuracy. As expected, very shallow trees in the ensemble underfit the data, which results in low accuracy on both training and test sets. As depth increases, accuracy improves significantly, indicating that the model is capturing more complex patterns in the data. However, beyond a certain depth, for example, depth 8 for Simple, 15 for Medium, and 20 for Complex, the test accuracy plateaus. The behavior is expected as once each tree has reached its maximum depth, growing the trees further does not improve performance. The ensemble itself also does not result in worse performance despite the trees being further overfitted, as the aggregating effect of the ensemble removes the noise. The optimal depths identified align with the complexity of each dataset, with more complex datasets requiring deeper trees to capture the structure. These results occur due to the bias-variance trade-off inherent in decision trees. Shallow trees have high bias and low variance, leading to underfitting, while deep trees have low bias and high variance, leading to overfitting. With more overfitted trees in the ensemble, the variance would average out more effectively, allowing for deeper trees to be used without overfitting.

B. Experiment 2: Number of Features per Split

Varying the number of features considered per split yields very modest gains that plateau. The inclusion of more features strengthens individual trees but increases inter-tree correlation, which limits bagging's variance reduction. Accuracy typically improves up to a mid-range, often near $\sqrt{d}$, and then remains flat or slightly declines, where d denotes the number of features. CV and test trends align, indicating limited sensitivity once depth is near-optimal.



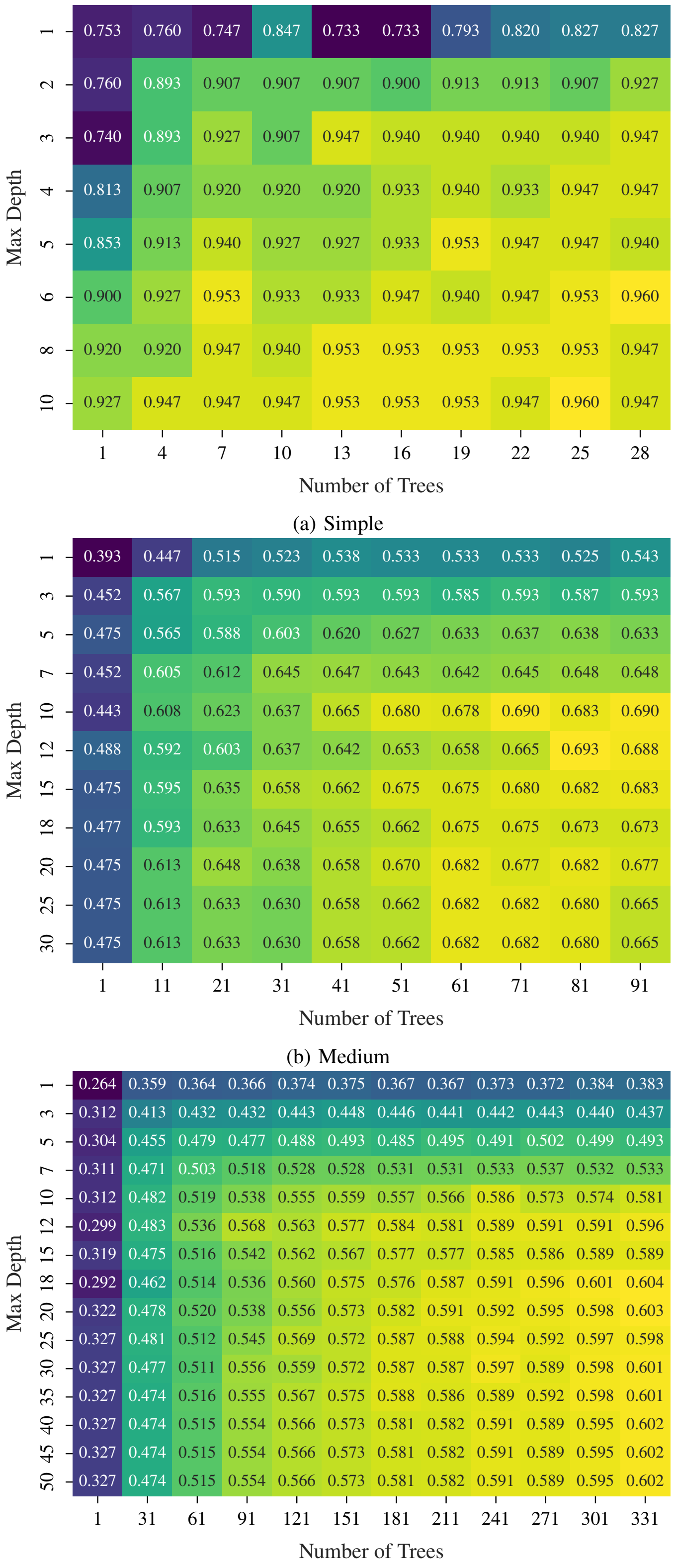
(a) Medium: Test and cross-validated accuracy versus maximum features considered per split.

Fig. 2: Number of features per split versus accuracy.

The results in Figure 2a illustrate how varying the number of features considered at each split affects test and cross-validated accuracy. As the number of features increases, there is only a slight improvement in accuracy up to a point, after which the performance does not show any clear signs of improvement. Because each individual tree was allowed to grow deep, the benefit of considering more features at each split is mitigated by the increased correlation between trees. The correlation reduces the effectiveness of bagging, as the ensemble relies on diverse predictions from its members to reduce variance. Therefore, while increasing the number of features per split enhances the strength of individual trees, the increased correlation also increases similarity, which limits the ensemble's overall performance. The results in the graph suggest that the optimal number of features to consider at each split is near the proposed heuristic of the square root of the total number of features, which is approximately three, five, and seven for Simple, Medium, and Complex datasets, respectively. The finding aligns with common practices in random forest implementations, where using a subset of features helps maintain diversity among trees while still allowing them to be sufficiently strong.

C. *Experiment 3: Depth versus Number of Trees*

The heatmaps show a clear interaction between depth and ensemble size: deeper trees require more trees to stabilize variance, while extra trees are needed to achieve better performance. Accuracy concentrates along a ridge where both parameters are sufficient; increasing one without the other offers little benefit, marking natural plateaus. Each heatmap in Figure 3 has the same pattern - there is a clear relationship between the number of trees in the ensemble and the maximum depth of the trees in the forest. Each heatmap shows a convergence to an optimal configuration. The forest created is dependent on the maximum depth, as seen in Experiment 1, but the performance is also dependent on the number of trees. The results show that even when increasing the maximum depth, the best results come from increasing the number of trees in conjunction. The general trend shows that more trees in the ensemble and deeper trees both contribute to higher accuracy, which is expected in the case of bagging. The improvement in accuracy with more and deeper trees is due to the overall reduction in variance that bagging provides, where, although individual trees overfit, the ensemble averages out respective errors. However, if there are not enough trees in the ensemble of overfitted trees, the variance does not average out sufficiently, leading to the observed degradation in test performance.



(a) Simple

(b) Medium

(c) Complex

Fig. 3: Heatmaps of test accuracy over maximum tree depth versus number of trees for each dataset.

To standardize subsequent comparisons, the following dataset-specific settings were fixed based on the first two experiments and the Experiment 3 ridge regions for subsequent experiments:

- `optimal_estimators` = {`simple`: 30, `medium`: 90, `complex`: 330}
- `optimal_depths` = {`simple`: 8, `medium`: 10, `complex`: 20}
- `optimal_num_split` = {`simple`: 5, `medium`: 7, `complex`: 15}

D. Experiment 4: Mixed-Depth Ensembles

The tables compare ensembles mixing underfitted and overfitted trees. Across Simple, Medium, and Complex, the overfit-only composition attained the best test and CV accuracy, which indicated that under strong bagging, averaging many high-variance trees outperformed mixing capacities.

TABLE II: Experiment 4 Simple Dataset: Composition versus accuracy.

Composition	Train accuracy	Test accuracy	Cross-validated accuracy
Only Underfitted	0.991	0.940	0.943 ± 0.016
Majority Underfitted	0.991	0.933	0.954 ± 0.028
Balanced	1.000	0.960	0.946 ± 0.031
Majority Overfitted	1.000	0.940	0.963 ± 0.021
Only Overfitted	**1.000**	**0.960**	$\mathbf{0.966 \pm 0.025}$

TABLE III: Experiment 4 Medium Dataset: Composition versus accuracy.

Composition	Train accuracy	Test accuracy	Cross-validated accuracy
Only Underfitted	0.969	0.655	0.669 ± 0.025
Majority Underfitted	0.984	0.677	0.661 ± 0.028
Balanced	1.000	0.660	0.677 ± 0.019
Majority Overfitted	1.000	0.652	0.672 ± 0.026
Only Overfitted	**1.000**	**0.685**	$\mathbf{0.691 \pm 0.021}$

TABLE IV: Experiment 4 Complex Dataset: Composition versus accuracy.

Composition	Train accuracy	Test accuracy	Cross-validated accuracy
Only Underfitted	1.000	0.576	0.541 ± 0.013
Majority Underfitted	1.000	0.583	0.550 ± 0.015
Balanced	1.000	0.595	0.550 ± 0.026
Majority Overfitted	1.000	0.597	0.564 ± 0.021
Only Overfitted	**1.000**	**0.610**	$\mathbf{0.563 \pm 0.008}$

The results in Tables II, III, and IV show the performance of ensembles with different compositions of underfitted and overfitted trees across the three datasets. A clear trend emerged where ensembles composed entirely of overfitted trees consistently achieved the highest test accuracy and cross-validated accuracy. The trend suggested that, in the context of bagging, the high variance of overfitted trees was effectively averaged out by the ensemble, which led to better generalization. Mixed compositions did not outperform the all-overfit configuration, which indicated that the benefits of averaging high-variance learners outweighed the potential advantages of bias-variance blending through mixed depths. The results are not surprising, but one might have guessed that a mix of underfitted and overfitted trees would provide a better balance of bias and variance, leading to improved generalization. However, the effectiveness of bagging in reducing variance appeared to be sufficient to allow overfitted trees to perform best. Therefore, including enough trees within the ensemble was essential to ensure robustness. In another approach, weaker learners could be used exclusively, such as shallow trees, and follow the boosting approach of sequentially correcting errors.

E. Global and post-hoc statistical tests

To test for differences among the five ensemble compositions on each dataset, a Friedman test was applied to 35 paired cross-validation blocks, followed by Wilcoxon signed-rank tests, which were one-sided and Holm-corrected, comparing the "Only Overfitted" configuration against each alternative. The omnibus Friedman results were: *simple*: $\chi^2 = 21.122$; *medium*: $\chi^2 = 69.402$; *complex*: $\chi^2 = 82.083$. Each test had $p < 0.0001$. The low p-values indicate statistically significant differences among configurations.

On the *simple* dataset, "Only Overfitted" is significantly better than "Only Underfitted" after Holm correction, with Holm $p = 0.001989$, while gains over the other three compositions are not statistically significant where mean accuracy deltas are small with values between 0.1% to 1.5%. The other configurations are not significantly different from each other because the problem is small and relatively easy. Trees saturate quickly, and the structures become highly similar across compositions. Strong bagging further reduces variance, so fold-wise paired differences fall within sampling noise. Refer to Table V for details.

TABLE V: Post-hoc Wilcoxon signed-rank test versus "Only Overfitted" - *simple*. One-sided alternative tests whether the reference outperforms the comparator with Holm-adjusted p-values controlling the family-wise error rate.

Comparator	W	p	Holm p	Reject	Mean Δ Accuracy
Only Underfitted	304.00	0.00050	0.00199	True	0.01470
Majority Underfitted	153.50	0.03428	0.10280	False	0.00730
Balanced	243.50	0.04017	0.10280	False	0.00570
Majority Overfitted	179.00	0.46420	0.46420	False	0.00120

On the *medium* dataset, "Only Overfitted" significantly outperforms all four alternatives under the one-sided Wilcoxon with Holm adjustment, with Holm $p < 0.004$, with effect sizes from 0.9% to 3.5%, indicating a robust advantage as task complexity increases. Refer to Table VI for the medium dataset details.

TABLE VI: Post-hoc Wilcoxon versus "Only Overfitted" - *medium*. One-sided alternative with Holm-adjusted p-values.

Comparator	W	p	Holm p	Reject	Mean Δ Accuracy
Only Underfitted	627.00	< 0.0001	< 0.0001	True	0.03520
Majority Underfitted	541.50	< 0.0001	< 0.0001	True	0.02130
Balanced	562.50	< 0.0001	< 0.0001	True	0.01710
Majority Overfitted	363.50	0.00349	0.00349	True	0.00950

On the *complex* dataset, “Only Overfitted” attains the highest test accuracy as seen in Table IV and the highest cross-validated mean accuracy, where the CV accuracy was 0.563 ± 0.008, showing both superior hold-out performance and the most stable CV results among compositions. Furthermore, Wilcoxon post-hoc tests confirm these gains are statistically significant against all alternatives where the Holm-adjusted value was $p \leq 0.00040$, with mean accuracy improvements ranging from +0.0077 for “Majority Overfitted” to +0.0286 for “Only Underfitted”. Refer to Table VII for details.

TABLE VII: Post-hoc Wilcoxon versus “Only Overfitted” - *complex*. One-sided alternative; Holm-adjusted p-values.

Comparator	W	p	Holm p	Reject	Mean Δ Accuracy
Only Underfitted	595.00	< 0.0001	< 0.0001	True	0.02860
Majority Underfitted	610.50	< 0.0001	< 0.0001	True	0.01950
Balanced	581.00	< 0.0001	< 0.0001	True	0.01150
Majority Overfitted	519.50	0.00040	0.00040	True	0.00770

VI. Conclusion

The assignment investigated how random forest performance is controlled by maximum tree depth, the number of features considered per split, and ensemble size across three datasets of increasing complexity. A reproducible pipeline was implemented with stratified splits, per-tree seeding, and five-fold cross-validation (CV), and four experiments were conducted: (i) depth sweeps, (ii) feature-subsampling per split sweeps at near-optimal depth, (iii) depth-by-trees interaction grids, and (iv) ensembles mixing under- and overfitting trees.

The main findings are consistent with bias-variance theory and answer the following:

- Maximum depth: Test and cross-validated accuracy increase from shallow to moderate depths, then plateau once the trees saturate. The depth that maximizes generalization increases with task complexity. With sufficiently many trees, deeper models become more viable because bagging reduces variance.
- Features per split: Allowing more features strengthens individual trees but raises inter-tree correlation. Accuracy improves to a plateau, after which gains are marginal or flat. The optimal number of features per split is often near $\sqrt{d}$.
- Depth versus ensemble size: The trend for better performance includes both depth and the number of trees being sufficiently large and tuned to the dataset complexity. Deeper trees require larger ensembles to stabilize variance. The parameters are limited by each other, where increasing one without the other beyond a certain point yields little benefit.
- Mixed-depth ensembles: Under strong bagging, ensembles composed solely of overfitted trees matched or outperformed mixed-capacity ensembles on test and CV accuracy, with comparable or smaller gaps.

The following practical suggestions emerged from these results: (i) pick depth by maximizing CV performance; (ii) grow the number of trees until CV mean stabilizes and its variance shrinks, noting that deeper forests require more trees; and (iii) use sufficiently many trees in ensembles of overfitted trees to obtain the best performance.

Limitations and future work: the study used synthetic, balanced datasets and focused on accuracy, where, if extending to real-world datasets, additional metrics and class imbalance would broaden conclusions. Secondly, exploring other ensemble methods like boosting or stacking could reveal different dynamics. Finally, investigating interactions with other hyperparameters, such as minimum samples per leaf or pruning, would provide a more comprehensive understanding of random forest tuning. Overall, the best configurations balance depth, feature subsampling, and ensemble size, with preferred values increasing with task complexity. Under strong bagging, larger ensembles of deeper trees generally deliver the most stable and accurate performance.

The code for the report is available at github.com/grantxxcoder/25849646rw441Assignment4.

Appendix

List of Acronyms

Acronym	**Definition**
CV	Cross-validation

References

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.