UWA

PERTH · AUSTRALIA

2024 Sem 1

CITS5508 Machine Learning

Assignment 2

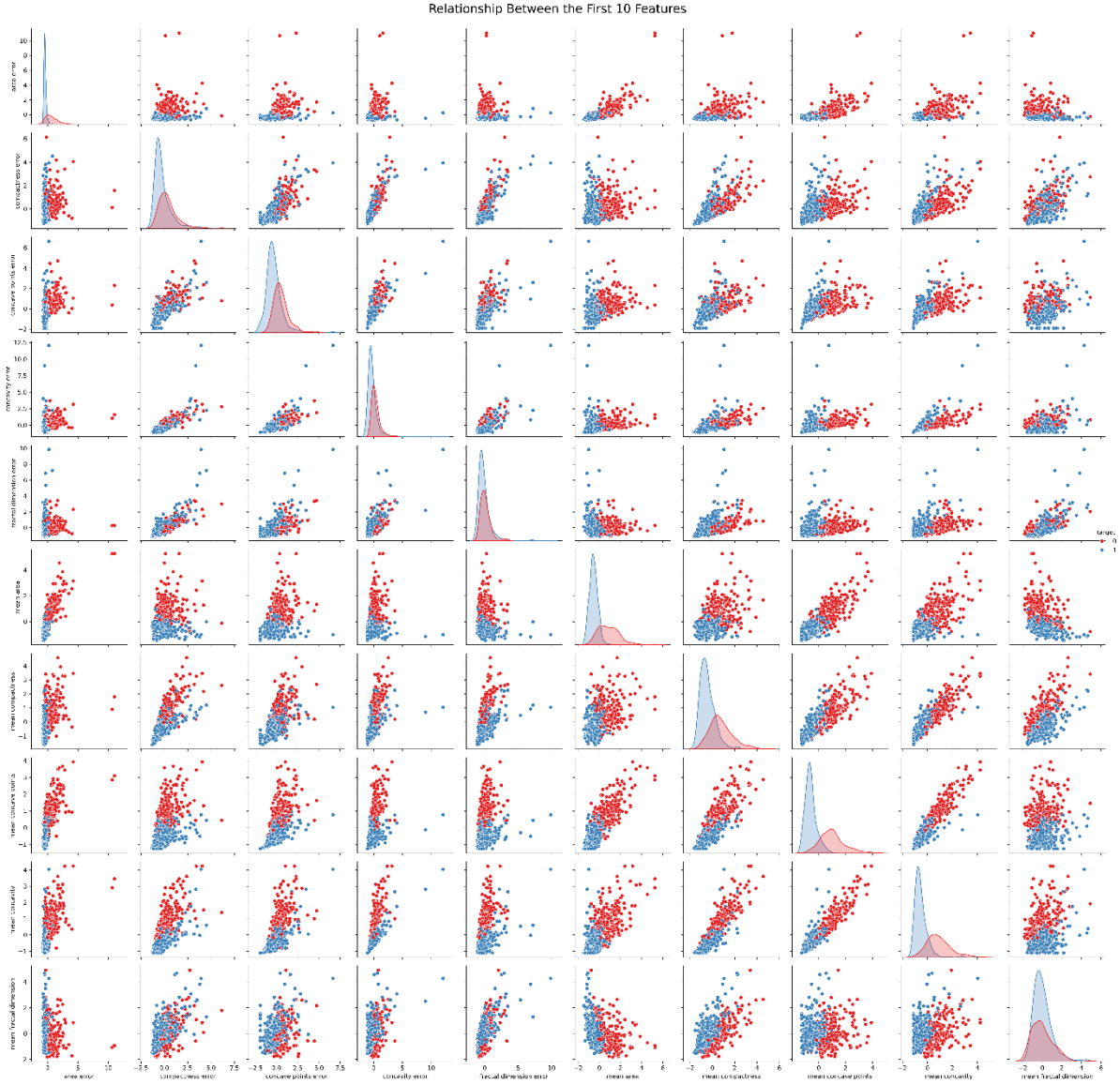
Mila Zhang (22756463)

Due date: Friday, 3 May, 8pm

Table of Contents

1. Initial Inspections and Data Preprocessing

1.1 Task D1



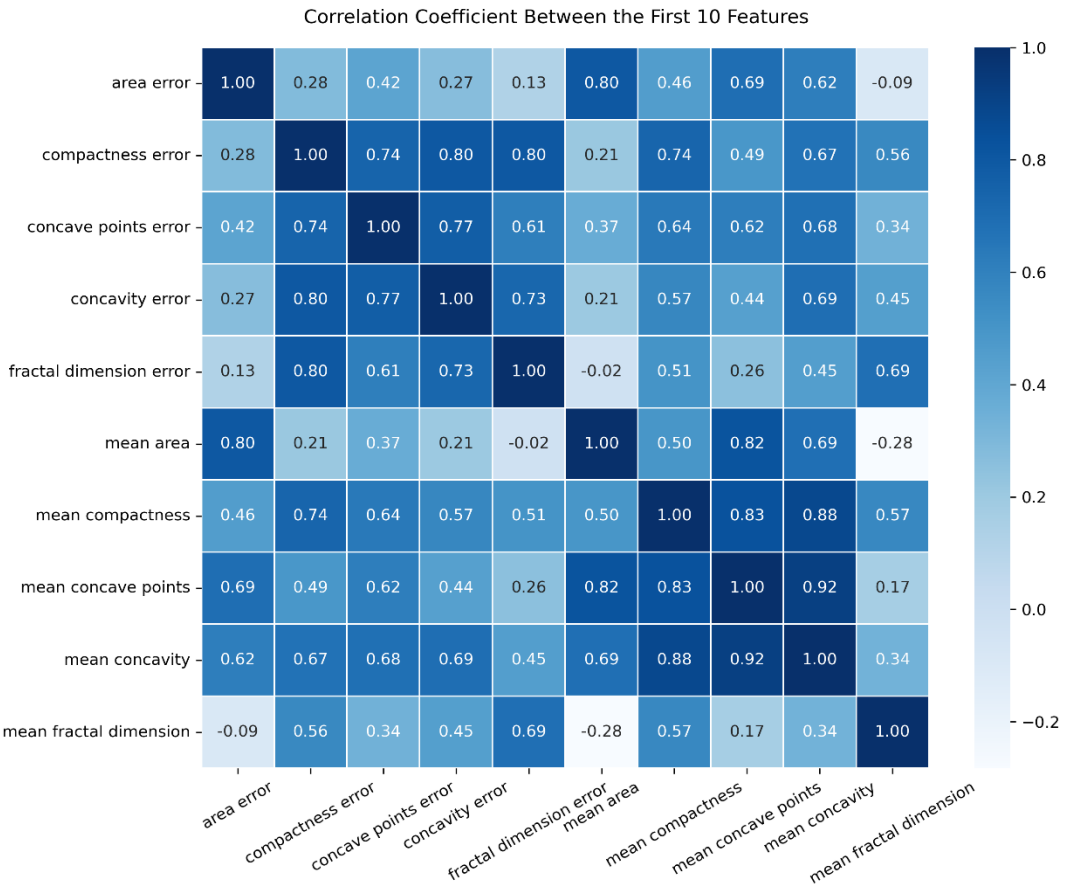
1.2 Task D2

Some pairs of features are *highly correlated* and the distribution of data is almost linear, while most pairs of features do not exhibit multicollinearity.

Some clusters of data points are well *separated* regarding different target labels, while some clusters are mixed up and cannot be separated well. Instances that are located far away from the clusters might be outliers.

For features that are highly correlated, one of them in each pair can be removed because they have a strong relationship with each other, and will essentially provide *redundant information* for splitting the data.

1.3 Task D3



1.4 Task D4

The correlation coefficients support the previous observations. The pairs of features that exhibit a *linear distribution* have *high correlation coefficient* values above 0.88, corresponding to the blocks with darker shades in the correlation matrix.

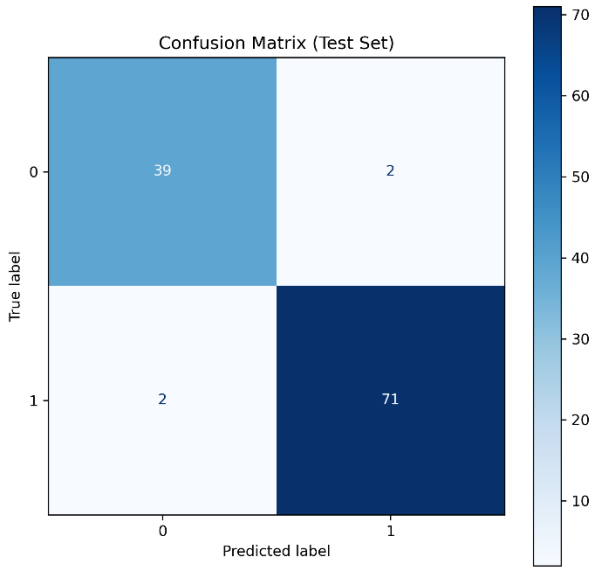
Assumption: Models are supposed to be trained using the reordered data (as done in D1).

2. Decision Tree Models with Default Hyperparameters

2.1 Task D6

Decision Tree with Default Hyperparameters

	Accuracy	Precision	Recall
Train	1.0	1.0	1.0
Test	0.96	0.97	0.97

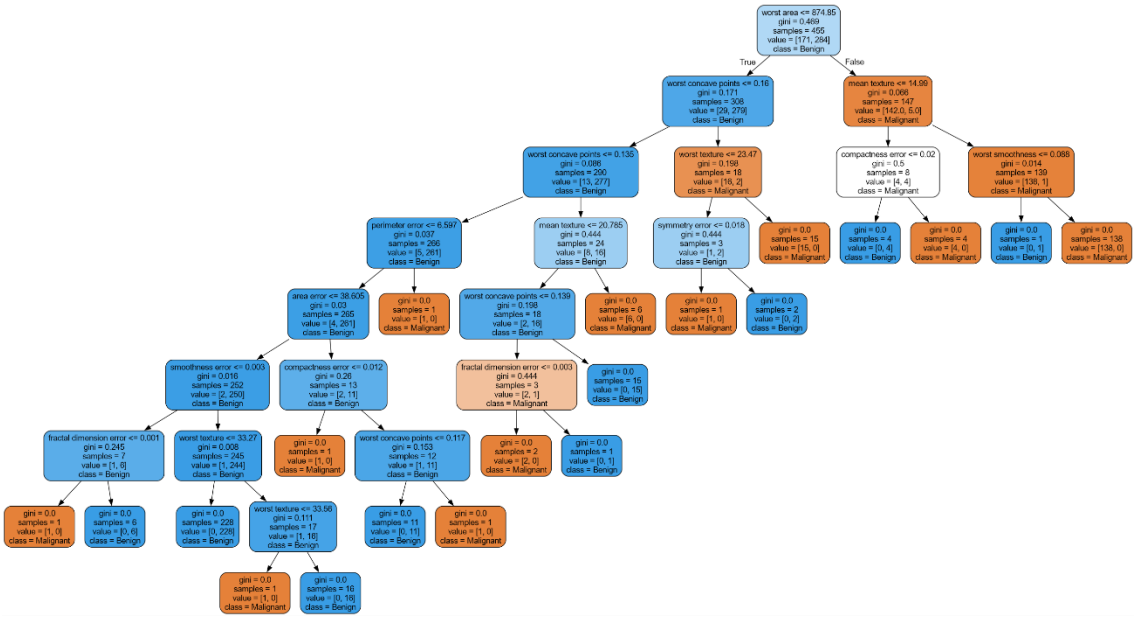


2.2 Task D7

The classifier, when configured with default hyperparameters is *overfitting*. This overfitting is evidenced by the extremely high scores (1.0) of the training set. The overfitting is also evidenced by the discrepancy in performance between the training and test datasets (e.g. the accuracy of the training set is 1.0 while that of the test set is just 0.96).

Without fine-turning, this model split the training data recursively until each branch terminates in a leaf node that has a Gini impurity of 0, capturing all the noise and anomalies in the training data. These anomalies might not represent the actual trends or patterns in the overall data population, leading to poorer generalisation on new unseen test data.

2.3 Task D8



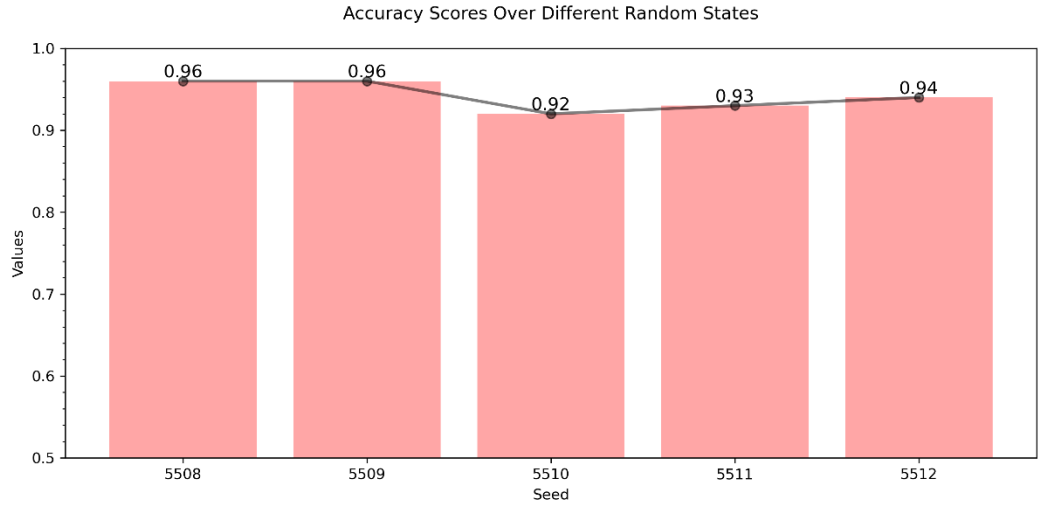
2.4 Task D9

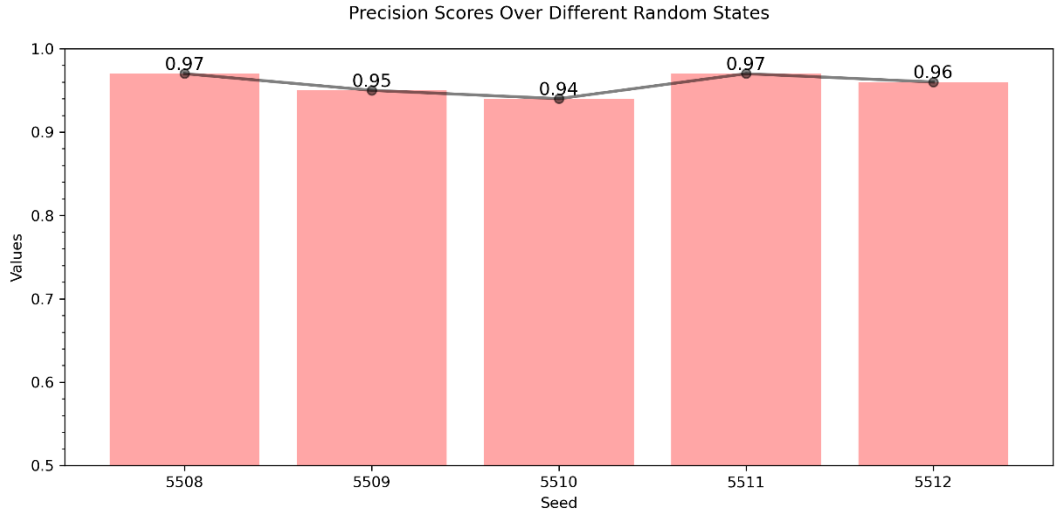
There are 9 levels in this decision tree model. This diagram helped to confirm the classifier has an overfitting issue.

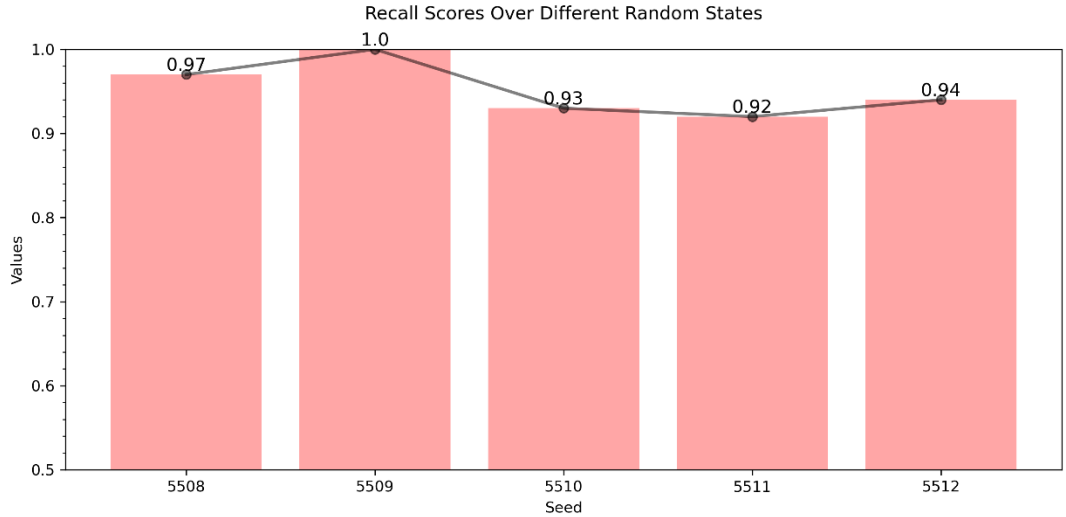
According to the diagram, it is observed that all samples in each leaf node belong to the same class, and each leaf node has a Gini impurity of zero. Additionally, the size of leaf nodes is very small, with a minimum of 1.

This is an interpretable model because the cause of each decision is demonstrated clearly in the tree and thus provides a transparent and straightforward nature of the decision-making process.

2.5 Task D10


Accuracy Scores Over Different Random States
1.0
0.9
0.8
0.7
0.6
0.5
Values
0.96
0.96
0.92
0.93
0.94
5508
5509
5510
5511
5512
Seed


Precision Scores Over Different Random States
1.0
0.9
0.8
0.7
0.6
0.5
Values
0.97
0.95
0.94
0.97
0.96
5508
5509
5510
5511
5512
Seed

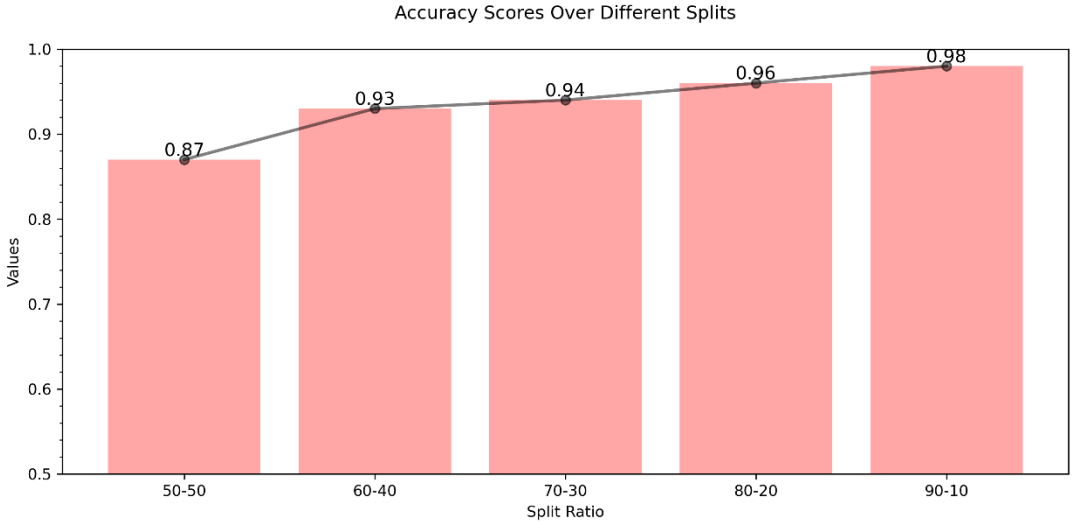

Recall Scores Over Different Random States
1.0
0.9
0.8
0.7
0.6
0.5
Values
0.97
1.0
0.93
0.92
0.94
5508
5509
5510
5511
5512
Seed

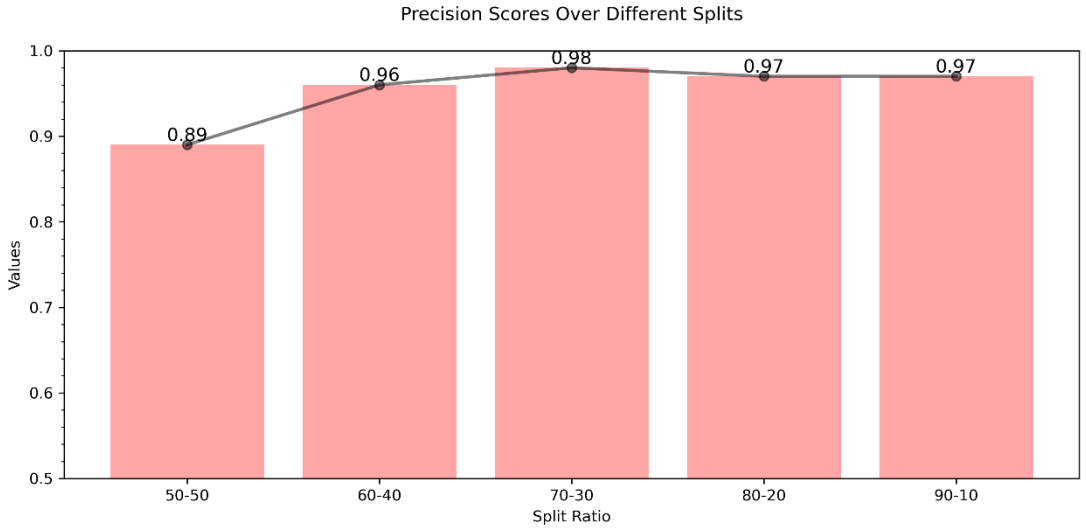
According to the figures, the scores range from 0.92 to 1.00, demonstrating high consistency with some variation (*fluctuation*) among the models with different random seeds.

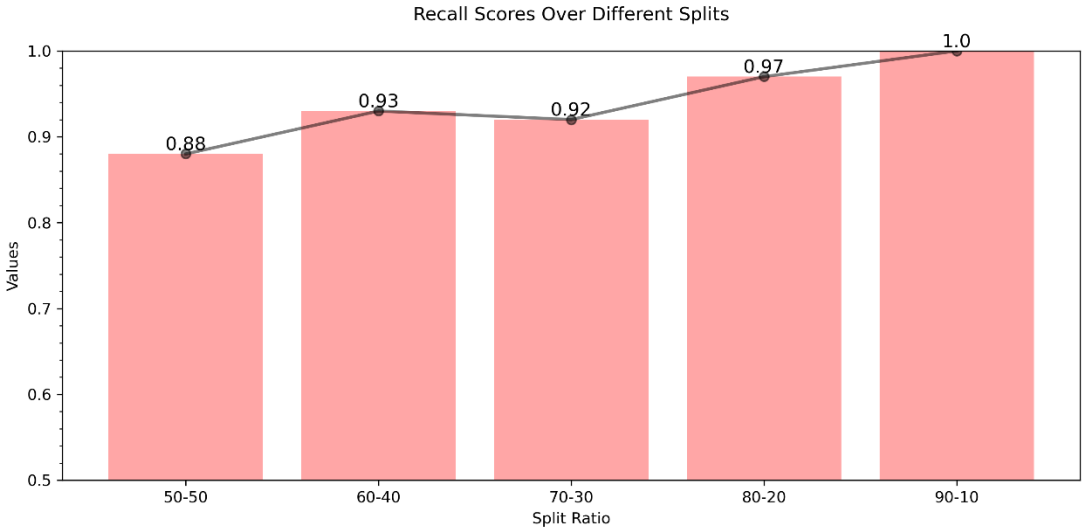
The observed variability results from these 5 models' inherent sensitivity to the *specific makeup of the training data* using different random seeds. Decision trees can produce different structures and therefore slightly different performance metrics depending on how the nodes are split at each decision point in the tree.

Despite this variability, the models demonstrate a high degree of consistency in performance, suggesting that the decision tree has captured the underlying patterns in the data effectively across different random seeds.

2.6 Task D11


Accuracy Scores Over Different Splits
Values
Split Ratio
50-50
60-40
70-30
80-20
90-10
0.87
0.93
0.94
0.96
0.98


Precision Scores Over Different Splits
Values
Split Ratio
50-50
60-40
70-30
80-20
90-10
0.89
0.96
0.98
0.97
0.97


Recall Scores Over Different Splits
Values
Split Ratio
50-50
60-40
70-30
80-20
90-10
0.88
0.93
0.92
0.97
1.0

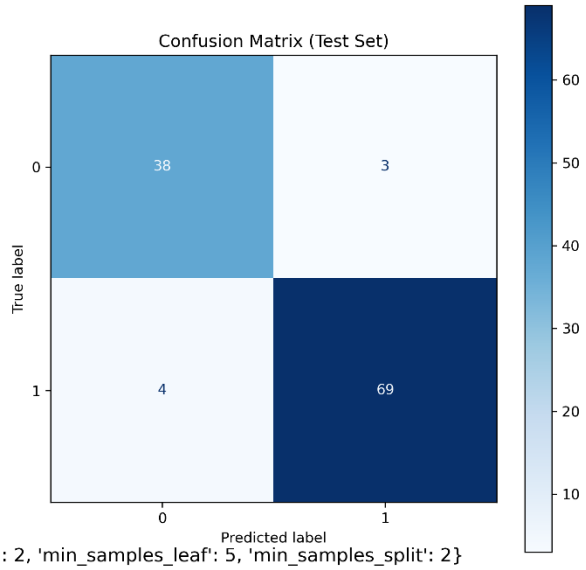
The scores across different split ratios exhibit a consistent trend of *improvement* as the proportion of the training set increases. This performance aligns with expectations, as when provided with more training data generally, the model has more training samples to learn from and is better trained, and thus perform and generalise better.

3. Decision Tree Models with Optimal Hyperparameters

3.1 Task D12

Evaluation Metrics (with the Optimal Hyperparameters, scoring='accuracy')

	Accuracy	Precision	Recall
Train	0.96	0.96	0.98
Test	0.94	0.96	0.95



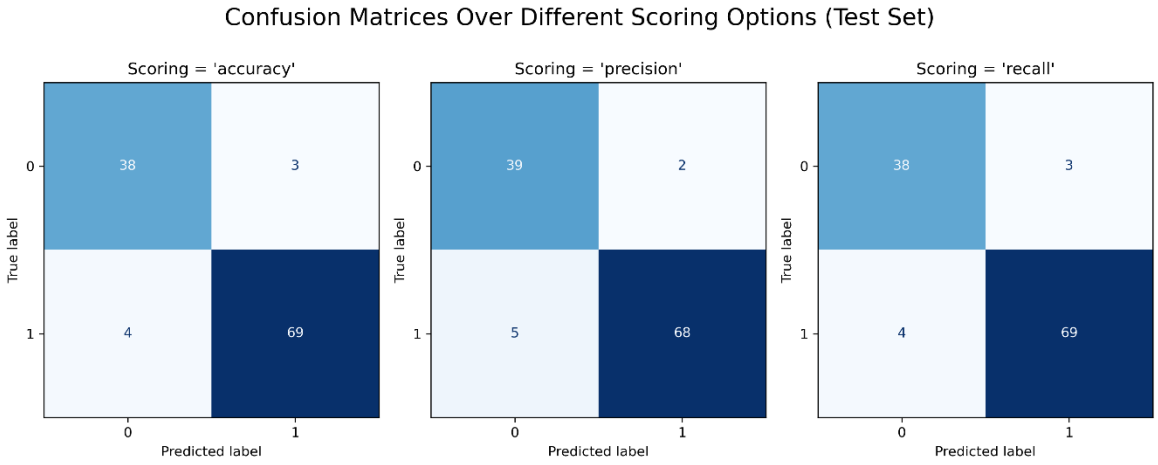
The optimal hyperparameters are: {'max_depth': 2, 'min_samples_leaf': 5, 'min_samples_split': 2}

3.2 Task D13

After fine-tuning the hyperparameters, the issue of overfitting is addressed and the scores of the training set are *no longer 1.0* as obtained in D6. And now the *discrepancies* in scores between the training set and the test set *are smaller* compared to those of the model without fine-tuning. This aligns with my expectations.

However, the scores (performance) of the fine-tuned model on the test set are lower than that of the model without fine-tuning. My guess is that the hyperparameter options we pass into the grid search are not the global optimal, so the optimal model selected by using these hyperparameters cannot outperform the original overfitting model.

3.3 Task D14



Optimal Hyperparameter\ Scoring Method	max_depth	min_samples_leaf	min_samples_split
accuracy	2	5	2
precision	5	2	2
recall	3	5	2

When using different scoring methods, the optimal hyperparameters are different, especially “max_depth” and “min_samples_leaf”. When hyperparameters are tuned to optimise different scoring metrics, they adjust the model to emphasise different aspects of the data.

The confusion matrices resemble each other which indicate similar model performance on the test set. This suggests that the models’ predictions are *robust* across various decision boundaries imposed by the hyperparameters. This similarity can indicate that the model is consistently identifying true positives, true negatives, false positives, and false negatives, regardless of the scoring method chosen during tuning.

4. Decision Tree Models with a Reduced Feature Set

4.1 Task D15

Feature Importances (Descending Order)

worst area	0.84
worst concave points	0.13
mean smoothness	0.02
area error	0.0
compactness error	0.0
concave points error	0.0
concavity error	0.0
fractal dimension error	0.0
mean area	0.0
mean compactness	0.0
mean concave points	0.0
mean concavity	0.0
mean fractal dimension	0.0
mean symmetry	0.0
mean texture	0.0
perimeter error	0.0
smoothness error	0.0
symmetry error	0.0
texture error	0.0
worst compactness	0.0
worst concavity	0.0
worst fractal dimension	0.0
worst smoothness	0.0
worst symmetry	0.0
worst texture	0.0

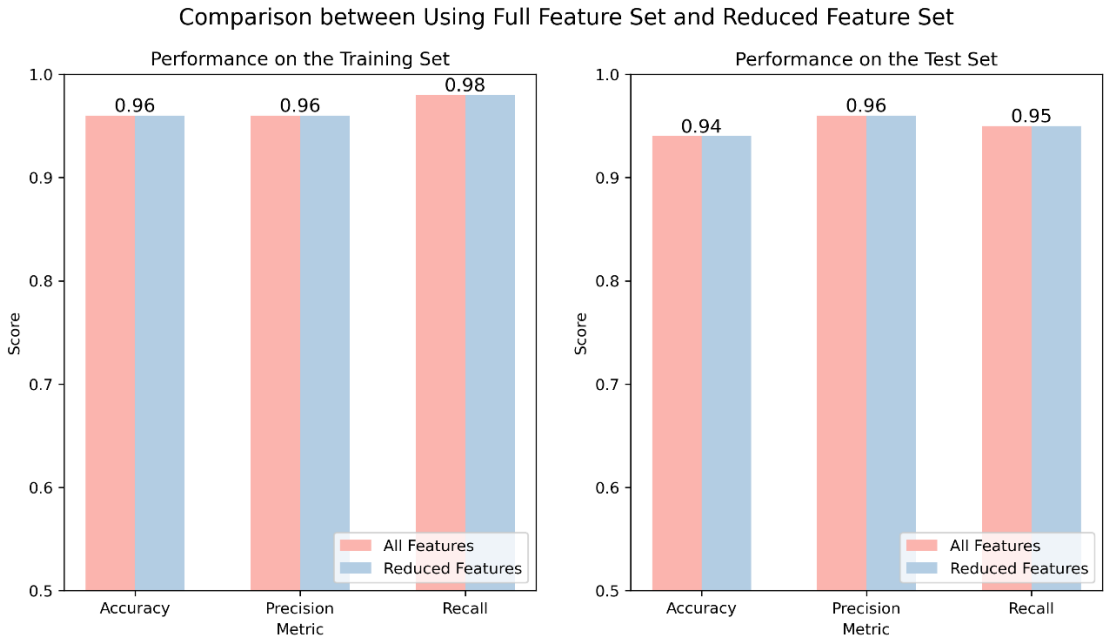
4.2 Task D16

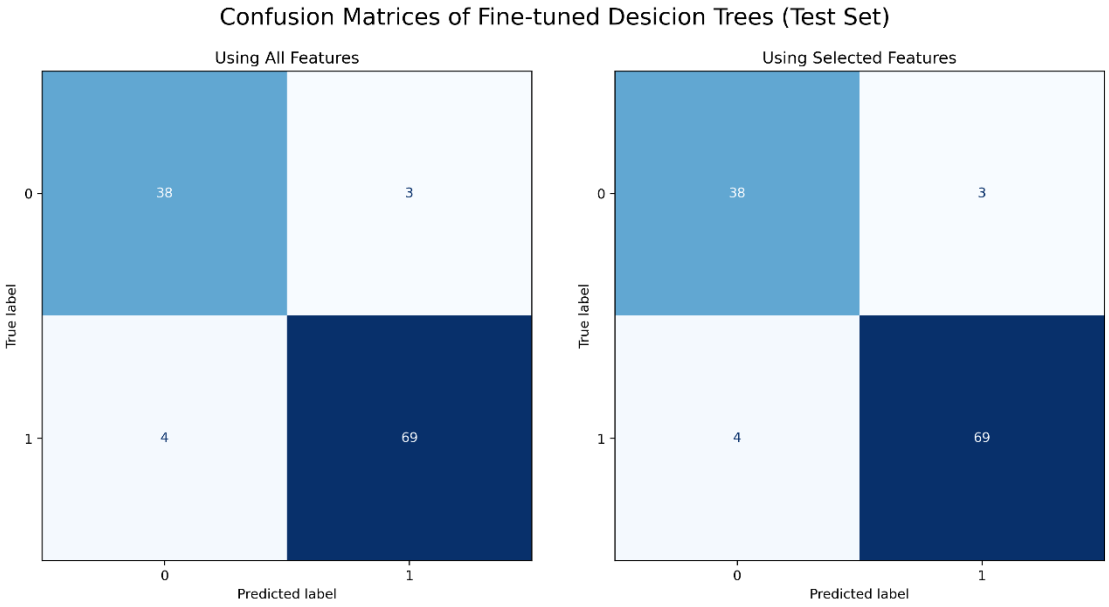
Retained features: ['mean smoothness', 'worst area', 'worst concave points']

Removed features: ['area error', 'compactness error', 'concave points error', 'concavity error', 'fractal dimension error', 'mean area', 'mean compactness', 'mean concave points', 'mean concavity', 'mean fractal dimension', 'mean symmetry', 'mean texture', 'perimeter error', 'smoothness error', 'symmetry error', 'texture error', 'worst compactness', 'worst concavity', 'worst fractal dimension', 'worst smoothness', 'worst symmetry', 'worst texture']

Total feature importance value after dimension reduction step: 1.0.

4.3 Task D17





4.4 Task D18

Reducing the number of features has *no impact* on the model performance. According to D15, those removed features all *have a feature importance of 0*, which means these features do not reduce impurity nor contribute to the prediction in each split.

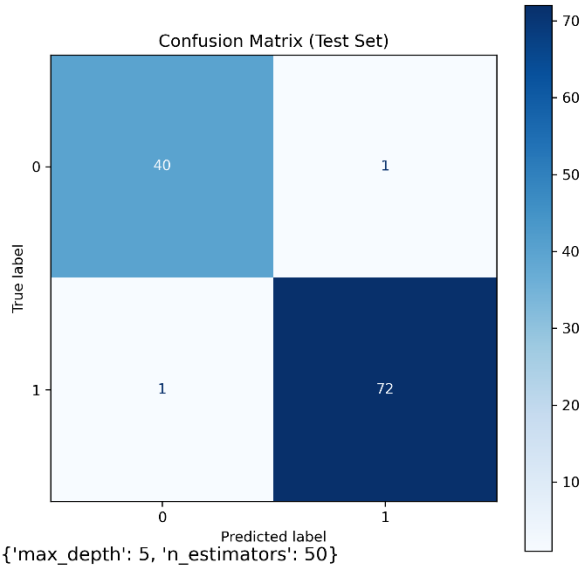
I did not repeat the cross-validation process to find the optimal hyperparameters when using the reduced set of features. The only variable is the feature set (whether reduced or not). This allows for the comparison between using the full feature set and the reduced feature set.

5. Random Forest Models

5.1 Task D19

Random Forest (with the Optimal Hyperparameters, scoring='accuracy')

	Accuracy	Precision	Recall
Train	0.99	0.99	1.0
Test	0.98	0.99	0.99



The optimal hyperparameters are: {'max_depth': 5, 'n_estimators': 50}

5.2 Task D20

The performance of the Random Forest Model is higher than the one obtained in D12. The scores of accuracy, precision, and recall are all higher than scored obtained in D12. The False-Negative and False-Positive misclassifications are fewer than that of the model in D12.

This result aligns with my expectations because Random Forests are an ensemble method that relies on the collective decision-making of multiple decision trees to improve accuracy and robustness. It effectively overcomes some of the limitations of individual decision trees and has higher overall performance than a single decision tree model.

5.3 Task D21

I do not believe these models are reliable enough for real-life application. They were trained on a small dataset. And through inspection in D1 we can observe anomalies and outliers existing in the dataset. These affects the prediction accuracy and its generalisation capabilities for unseen data.

A more complex model is not necessary given the high recall, accuracy, precision scores and good performance of confusion matrices, especially for the Random Forest model in D19.

At this stage, using a machine learning algorithm for this task is not advisable. That is, this decision process should not be automated. Identifying breast cancer is a crucial task and we do not want to misclassify any case. These models are more appropriate as tools to assist in manual diagnosis, rather than standalone diagnostic solutions.

The dataset used is a small dataset with less than 600 instances which is too small to build robust models. Also, there are anomalies and outliers observed in D1. It is better to collect more data and perform data cleaning before moving to the modelling stage.