2024 Sem 1

CITS5508 Machine Learning

Assignment 2

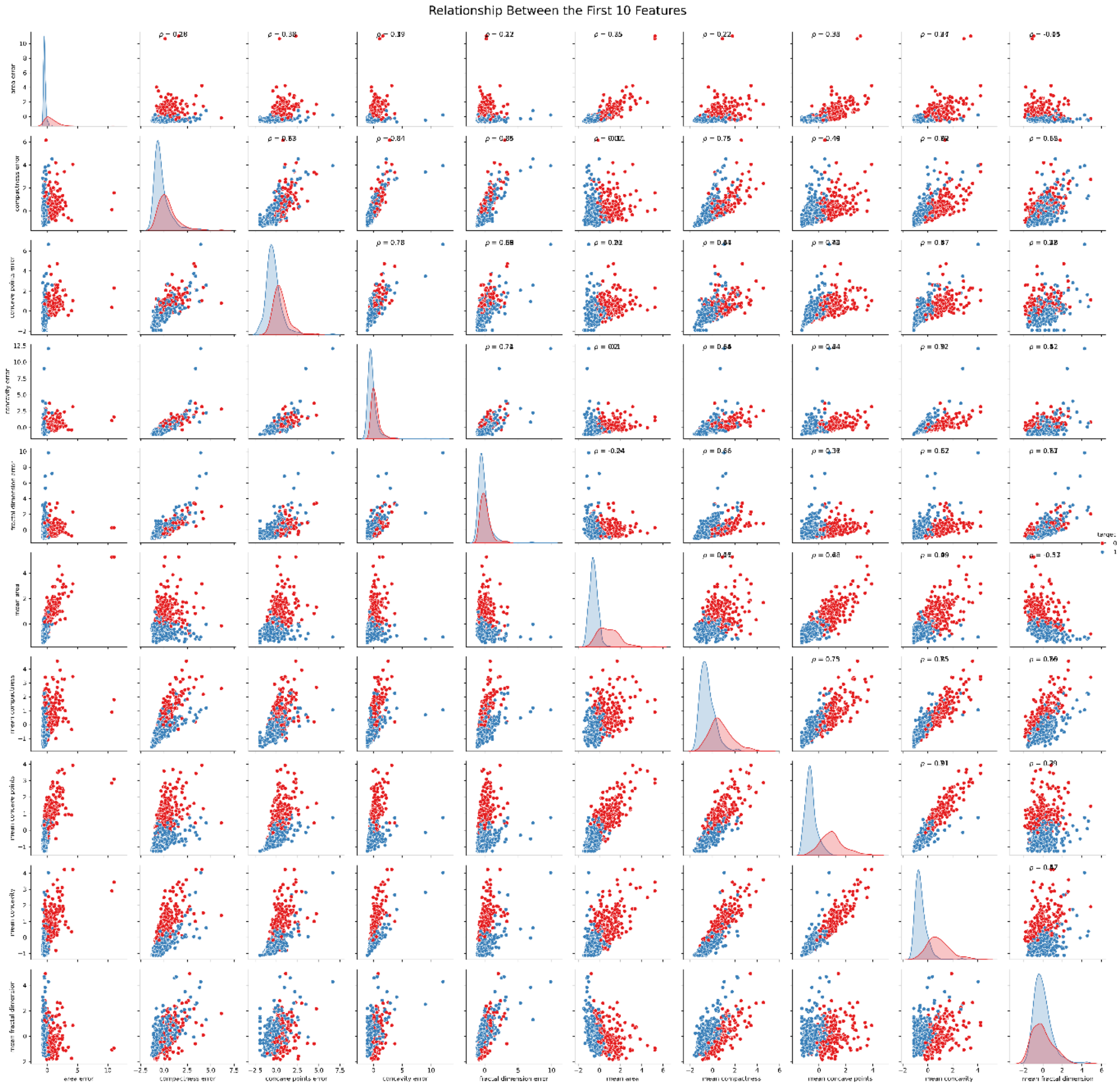
Mila Zhang (22756463)

Due date: Friday, 3 May, 8pm

Table of Contents

1. Initial Inspections and Data Preprocessing

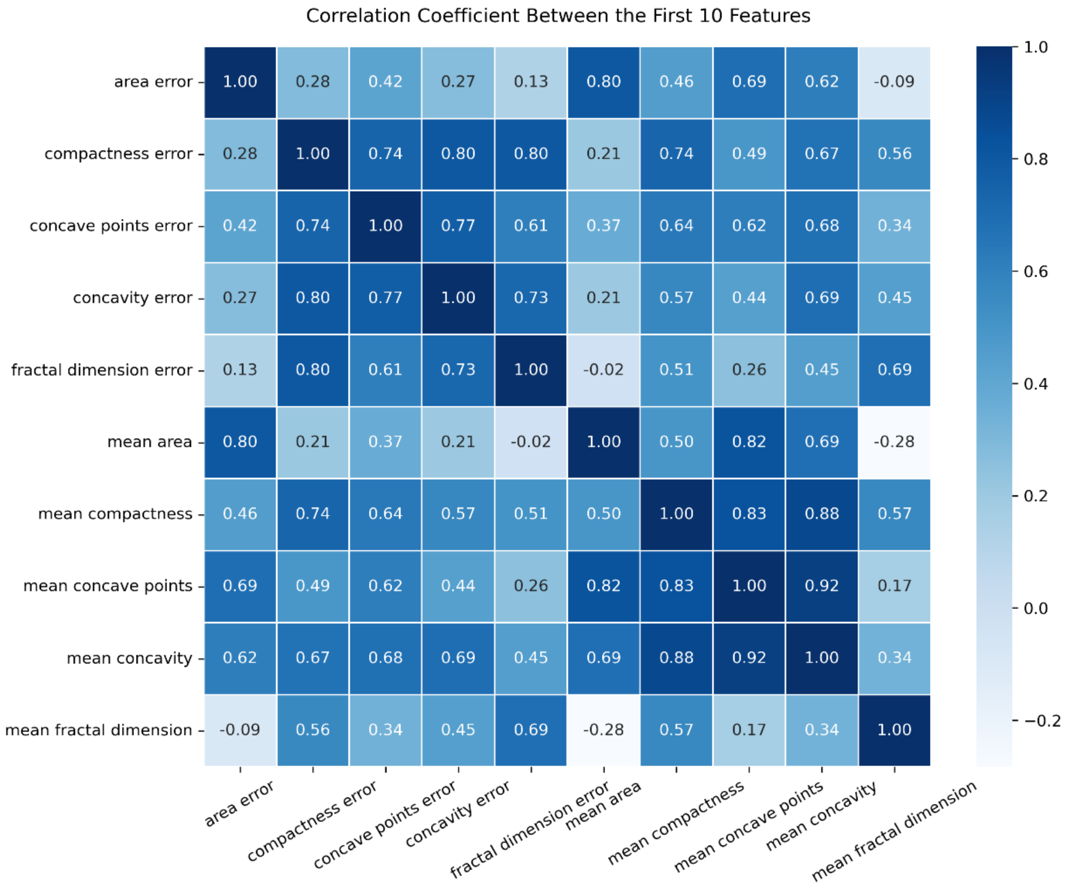
1.1 Task D1


Relationship Between the First 10 Features

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


Correlation Coefficient Between the First 10 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
area error	1.00	0.28	0.42	0.27	0.13	0.80	0.46	0.69	0.62	-0.09
compactness error	0.28	1.00	0.74	0.80	0.80	0.21	0.74	0.49	0.67	0.56
concave points error	0.42	0.74	1.00	0.77	0.61	0.37	0.64	0.62	0.68	0.34
concavity error	0.27	0.80	0.77	1.00	0.73	0.21	0.57	0.44	0.69	0.45
fractal dimension error	0.13	0.80	0.61	0.73	1.00	-0.02	0.51	0.26	0.45	0.69
mean area	0.80	0.21	0.37	0.21	-0.02	1.00	0.50	0.82	0.69	-0.28
mean compactness	0.46	0.74	0.64	0.57	0.51	0.50	1.00	0.83	0.88	0.57
mean concave points	0.69	0.49	0.62	0.44	0.26	0.82	0.83	1.00	0.92	0.17
mean concavity	0.62	0.67	0.68	0.69	0.45	0.69	0.88	0.92	1.00	0.34
mean fractal dimension	-0.09	0.56	0.34	0.45	0.69	-0.28	0.57	0.17	0.34	1.00

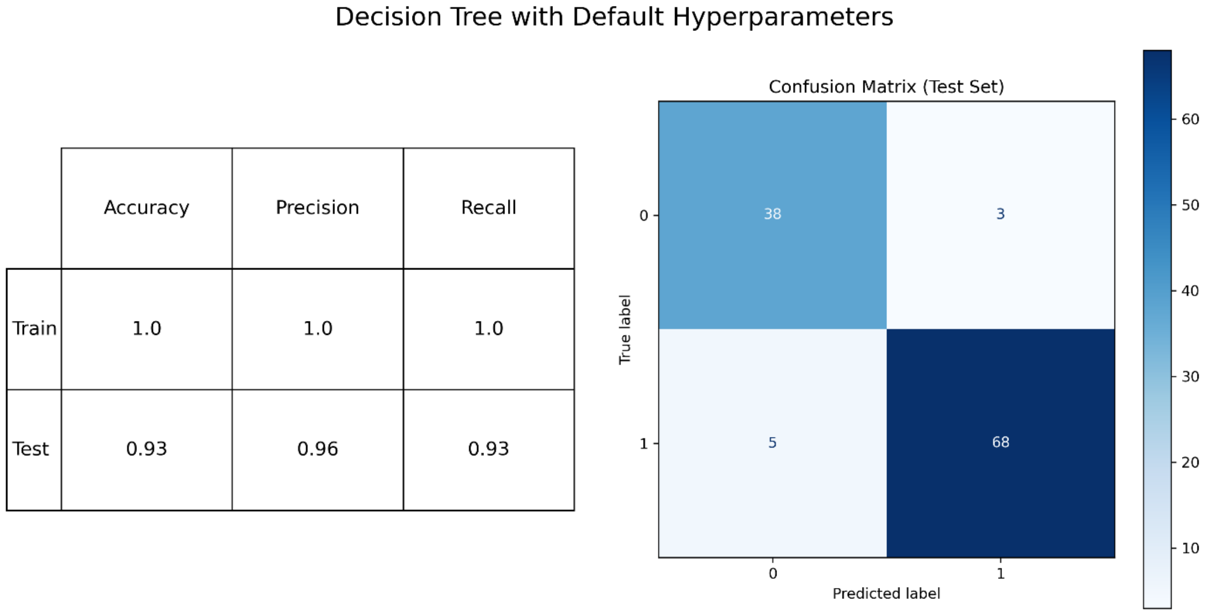
1.4 Task D4

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 unordered data. The column reordering in D1 is for inspecting the correlations between variables and does not alter the original dataset.

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	0.93	0.96	0.93

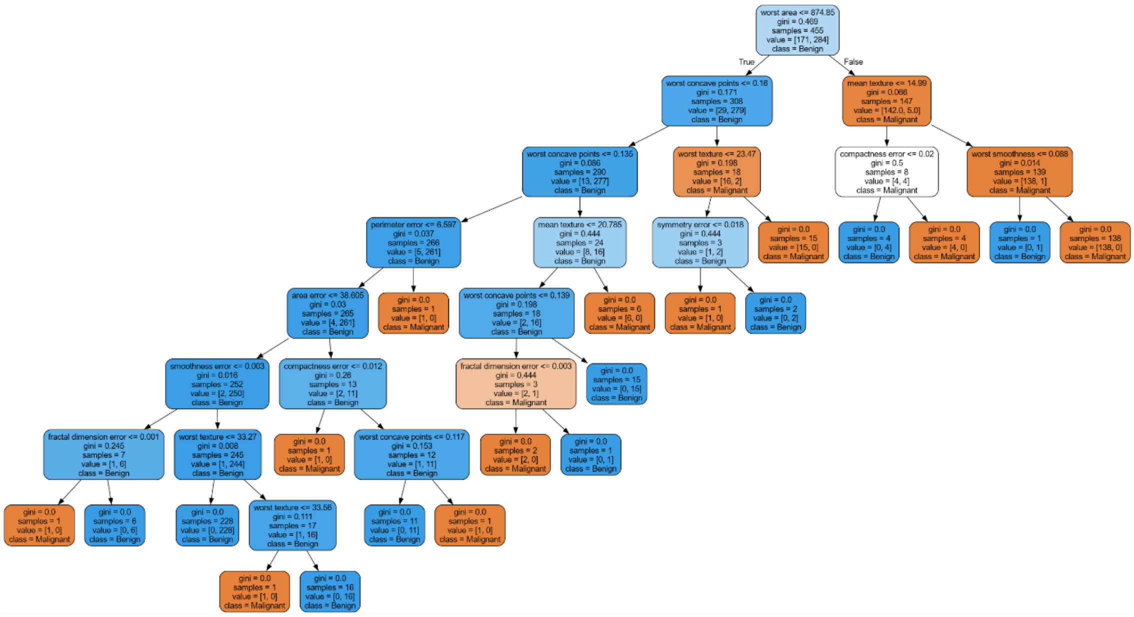
2.2 Task D7

The classifier, when configured with default hyperparameters is overfitting. This overfitting is evidenced by the significant 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 0.93).

Decision trees inherently operate by recursively splitting the data until each branch terminates in a leaf node, where either all data in the node belong to a single class or no further splits can reduce Gini impurity.

Relying on default hyperparameters, this method optimises performance exclusively for the training dataset and does not necessarily generalise well on unseen data. A fully-grown decision tree can perfectly classify all training data points, 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.

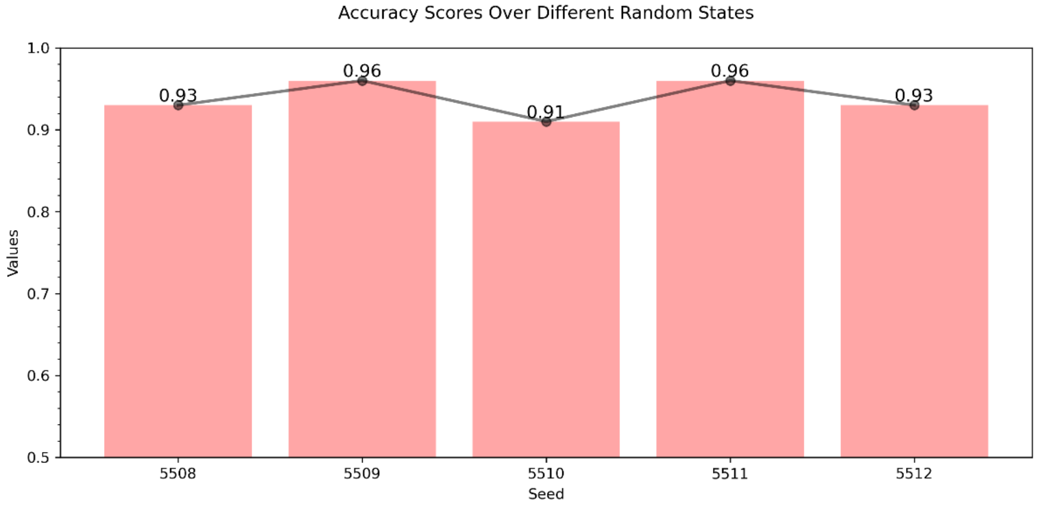
2.3 Task D8

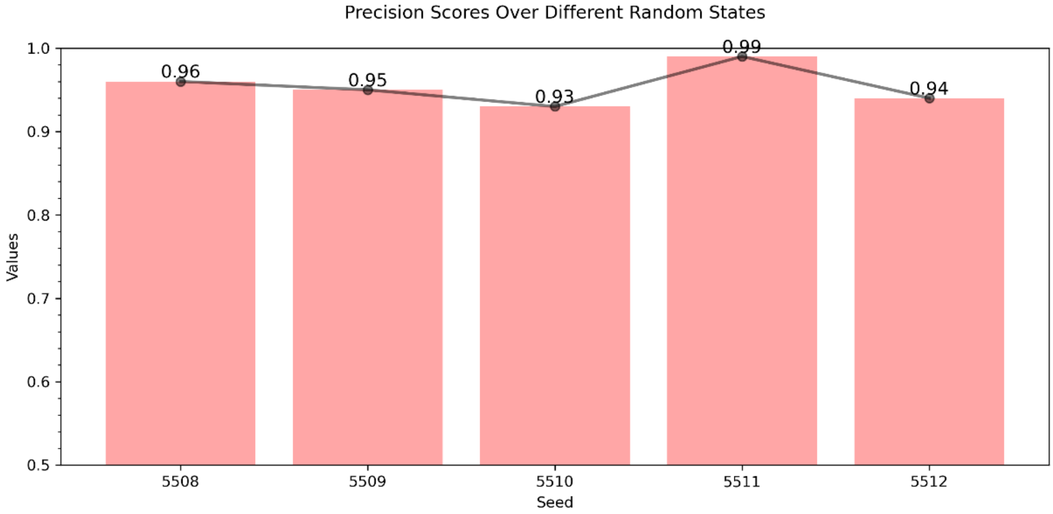

worst area <= 874.85 gini = 0.469 samples = 455 value = [171, 284] class = Benign
True
False
worst concave points <= 0.16 gini = 0.171 samples = 308 value = [29, 279] class = Benign
mean texture <= 14.99 gini = 0.066 samples = 147 value = [142.0, 5.0] class = Malignant
worst concave points <= 0.135 gini = 0.086 samples = 290 value = [13, 277] class = Benign
worst texture <= 23.47 gini = 0.198 samples = 18 value = [16, 2] class = Malignant
compactness error <= 0.02 gini = 0.5 samples = 8 value = [4, 4] class = Malignant
worst smoothness <= 0.088 gini = 0.014 samples = 139 value = [138, 1] class = Malignant
perimeter error <= 6.597 gini = 0.037 samples = 266 value = [5, 261] class = Benign
mean texture <= 20.785 gini = 0.444 samples = 24 value = [8, 16] class = Benign
symmetry error <= 0.018 gini = 0.444 samples = 3 value = [1, 2] class = Benign
gini = 0.0 samples = 15 value = [15, 0] class = Malignant
gini = 0.0 samples = 4 value = [0, 4] class = Benign
gini = 0.0 samples = 4 value = [4, 0] class = Malignant
gini = 0.0 samples = 1 value = [0, 1] class = Benign
gini = 0.0 samples = 138 value = [138, 0] class = Malignant
area error <= 38.605 gini = 0.03 samples = 265 value = [4, 261] class = Benign
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
worst concave points <= 0.139 gini = 0.198 samples = 18 value = [2, 16] class = Benign
gini = 0.0 samples = 6 value = [6, 0] class = Malignant
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
gini = 0.0 samples = 2 value = [0, 2] class = Benign
smoothness error <= 0.003 gini = 0.016 samples = 252 value = [2, 250] class = Benign
compactness error <= 0.012 gini = 0.26 samples = 13 value = [2, 11] class = Benign
fractal dimension error <= 0.003 gini = 0.444 samples = 3 value = [2, 1] class = Malignant
gini = 0.0 samples = 15 value = [0, 15] class = Benign
fractal dimension error <= 0.001 gini = 0.245 samples = 7 value = [1, 6] class = Benign
worst texture <= 33.27 gini = 0.008 samples = 245 value = [1, 244] class = Benign
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
worst concave points <= 0.117 gini = 0.153 samples = 12 value = [1, 11] class = Benign
gini = 0.0 samples = 2 value = [2, 0] class = Malignant
gini = 0.0 samples = 1 value = [0, 1] class = Benign
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
gini = 0.0 samples = 6 value = [0, 6] class = Benign
gini = 0.0 samples = 228 value = [0, 228] class = Benign
worst texture <= 33.56 gini = 0.111 samples = 17 value = [1, 16] class = Benign
gini = 0.0 samples = 11 value = [0, 11] class = Benign
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
gini = 0.0 samples = 1 value = [1, 0] class = Malignant
gini = 0.0 samples = 16 value = [0, 16] class = Benign

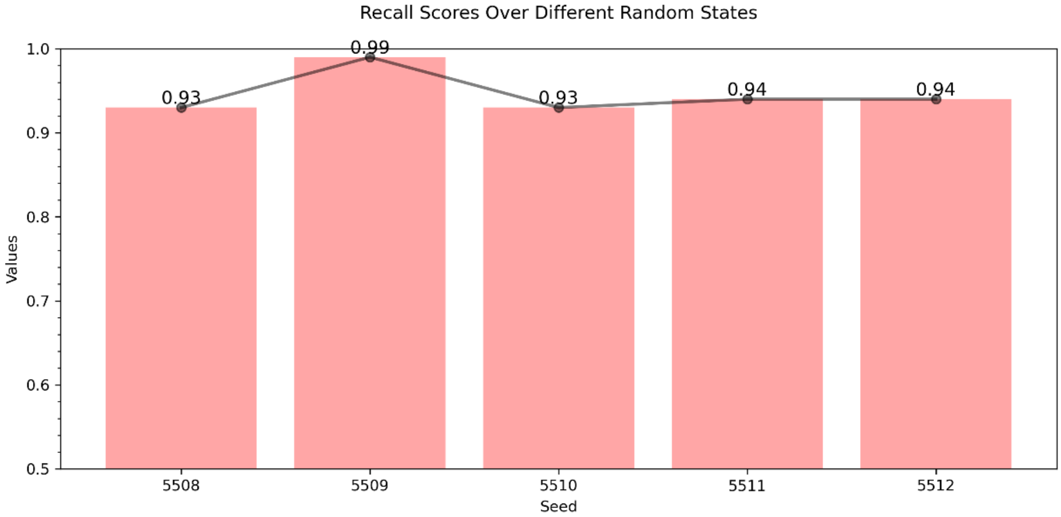
2.4 Task D9

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. 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
Values
1.0
0.9
0.8
0.7
0.6
0.5
0.93
0.96
0.91
0.96
0.93
5508
5509
5510
5511
5512
Seed


Precision Scores Over Different Random States
Values
1.0
0.9
0.8
0.7
0.6
0.5
0.96
0.95
0.93
0.99
0.94
5508
5509
5510
5511
5512
Seed

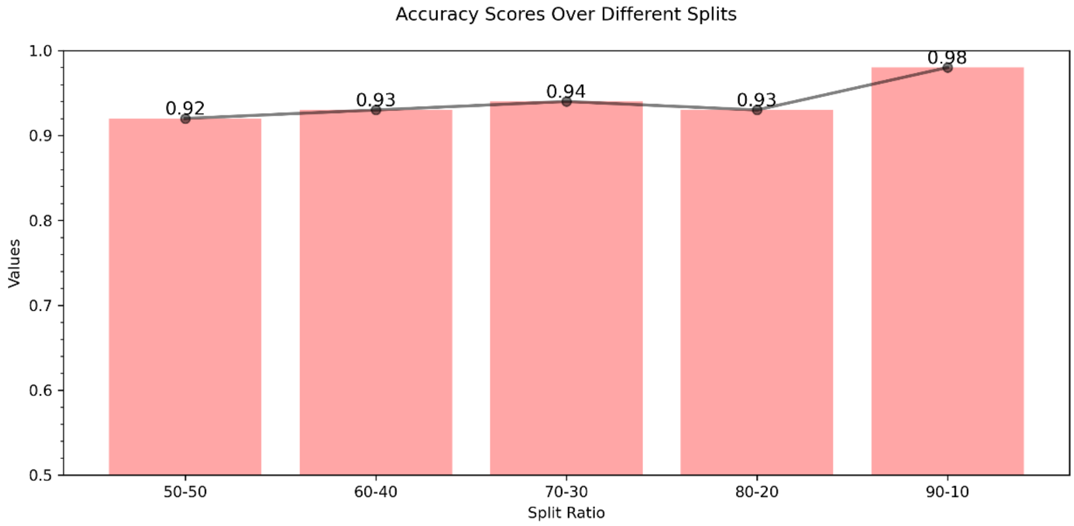

Recall Scores Over Different Random States
Values
1.0
0.9
0.8
0.7
0.6
0.5
0.93
0.99
0.93
0.94
0.94
5508
5509
5510
5511
5512
Seed

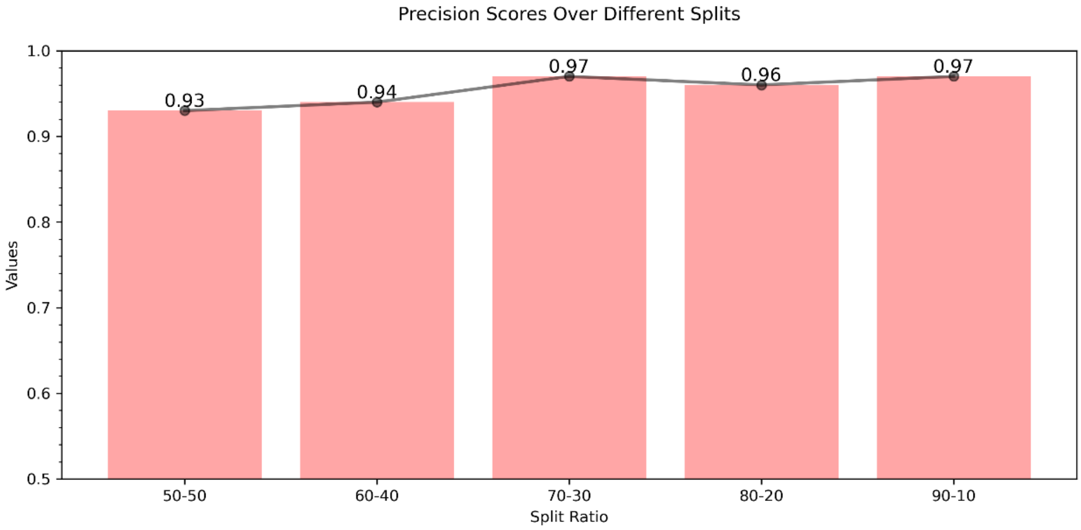
According to the figures, the scores range from 0.91 to 1.00, demonstrating high consistency with some variation among the different models.

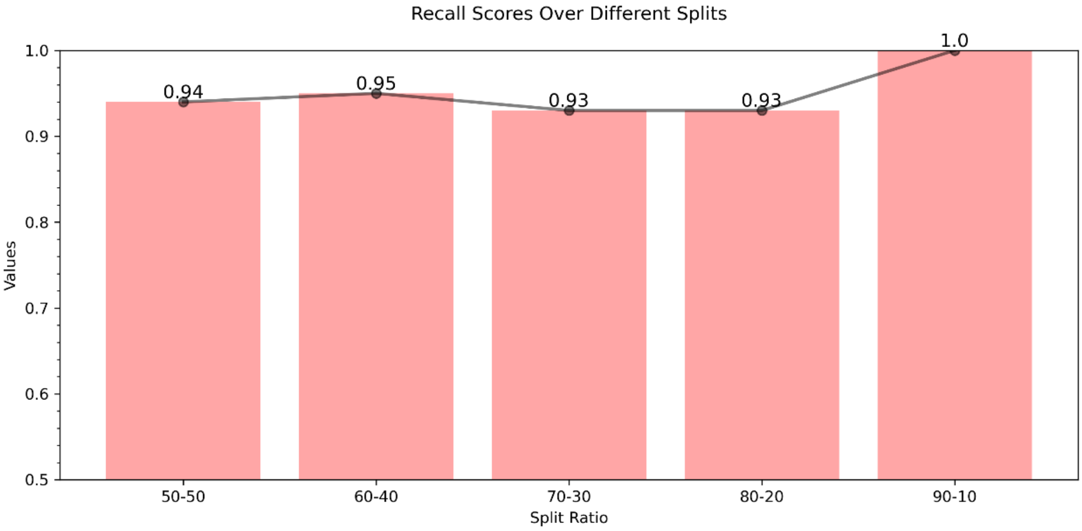
The observed variability across different runs with different seeds results from their inherent sensitivity to the specific makeup of the training data used. Decision trees can produce different structures and therefore slightly different performance metrics depending on how the nodes are split at each 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 all seeds.

2.6 Task D11


Accuracy Scores Over Different Splits
Values
Split Ratio
1.0
0.9
0.8
0.7
0.6
0.5
50-50
60-40
70-30
80-20
90-10
0.92
0.93
0.94
0.93
0.98

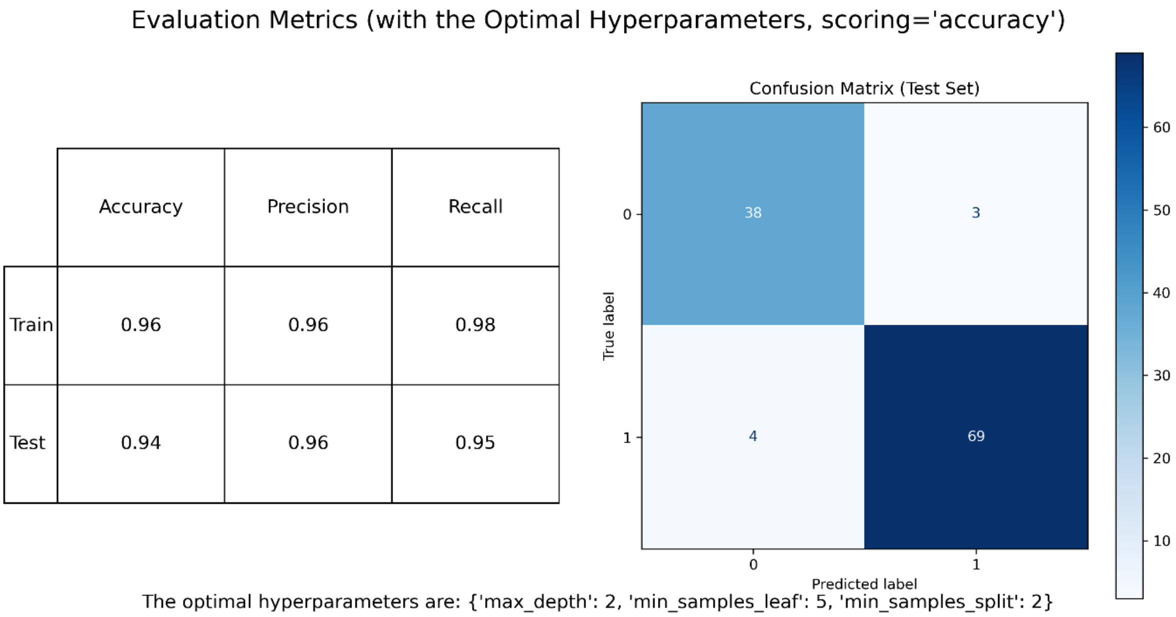

Precision Scores Over Different Splits
Values
Split Ratio
1.0
0.9
0.8
0.7
0.6
0.5
50-50
60-40
70-30
80-20
90-10
0.93
0.94
0.97
0.96
0.97


Recall Scores Over Different Splits
Values
Split Ratio
1.0
0.9
0.8
0.7
0.6
0.5
50-50
60-40
70-30
80-20
90-10
0.94
0.95
0.93
0.93
1.0

The accuracy and precision scores over different split ratios exhibit an overall tendency of increase as the training set ratio gets larger. This performance behaves as expected because the model is better trained when fed more training data generally. However, the recall scores increase then drop then increase again which is not as optimal as I expected.

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

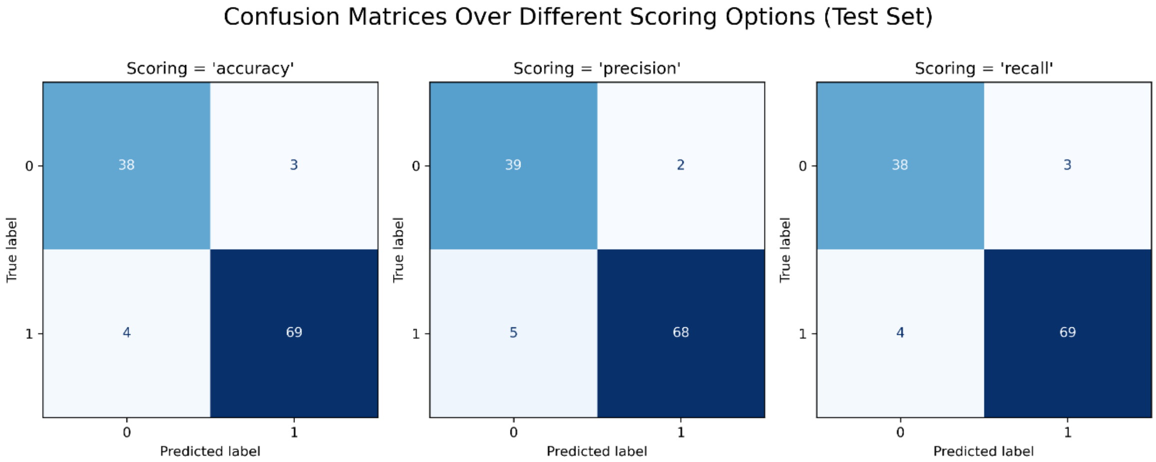
The optimal hyperparameters are: {'max_depth': 2, 'min_samples_leaf': 5, 'min_samples_split': 2}

3.2 Task D13

After fine-tuning the hyperparameters, the issue of overfitting is addressed and the model has better generalisation. This aligns with my expectations.

The scores of the training set are no longer 1.0 as obtained in D6, and the scores of the fine-tuned model on the test set are generally higher than that of the model without fine-tuning. Now the discrepancies in scores between the training set and the test set are smaller compared to those of the model without fine-tuning. These have shown that fine-tuning reduced the issue of overfitting in D6. However, after fine-tuning, the performance on test set is lower than the model without fine-tuning.

3.3 Task D14



Optimal Hyperparameter\ Scoring Method	max_depth	min_samples_leaf	min_samples_split
accuracy	4	5	2
precision	5	2	2
recall	3	5	2

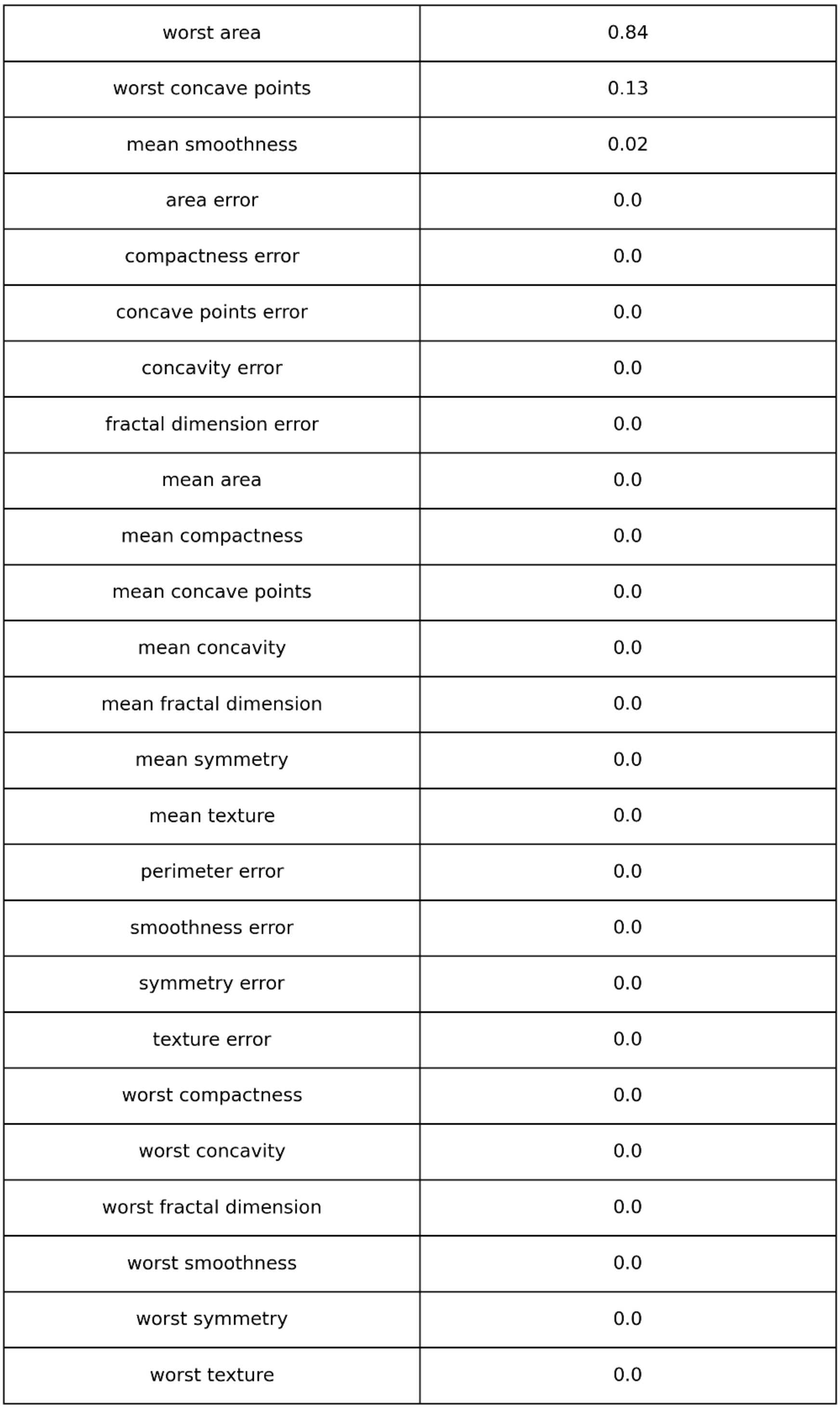
When using different scoring methods, the optimal hyperparameters are different, especially “max_depth” and “min_samples_leaf”. When hyperparameters are tuned to optimise different scoring metrics, they adjust the model to emphasise different aspects of the data.

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 hyperparameter settings prioritised 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

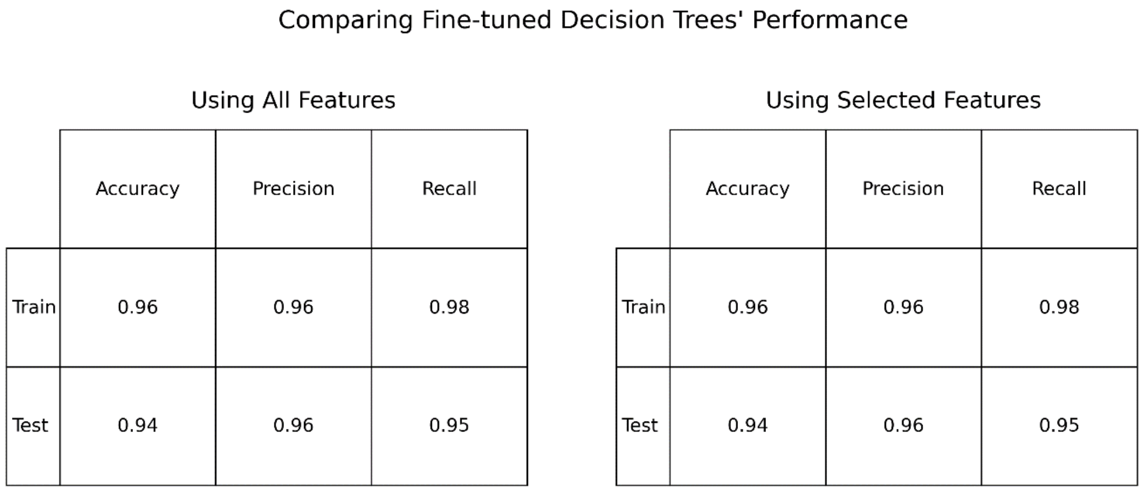
4.2 Task D16

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



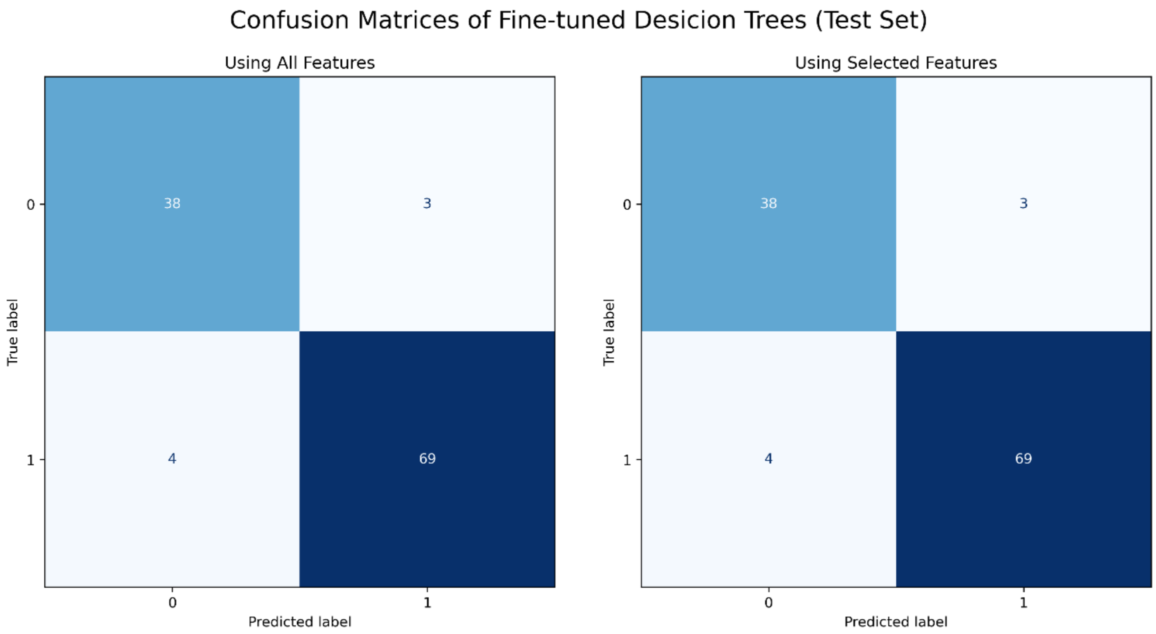
Comparing Fine-tuned Decision Trees' Performance

Using All Features

	Accuracy	Precision	Recall
Train	0.96	0.96	0.98
Test	0.94	0.96	0.95

Using Selected Features

	Accuracy	Precision	Recall
Train	0.96	0.96	0.98
Test	0.94	0.96	0.95



Confusion Matrices of Fine-tuned Desicion Trees (Test Set)

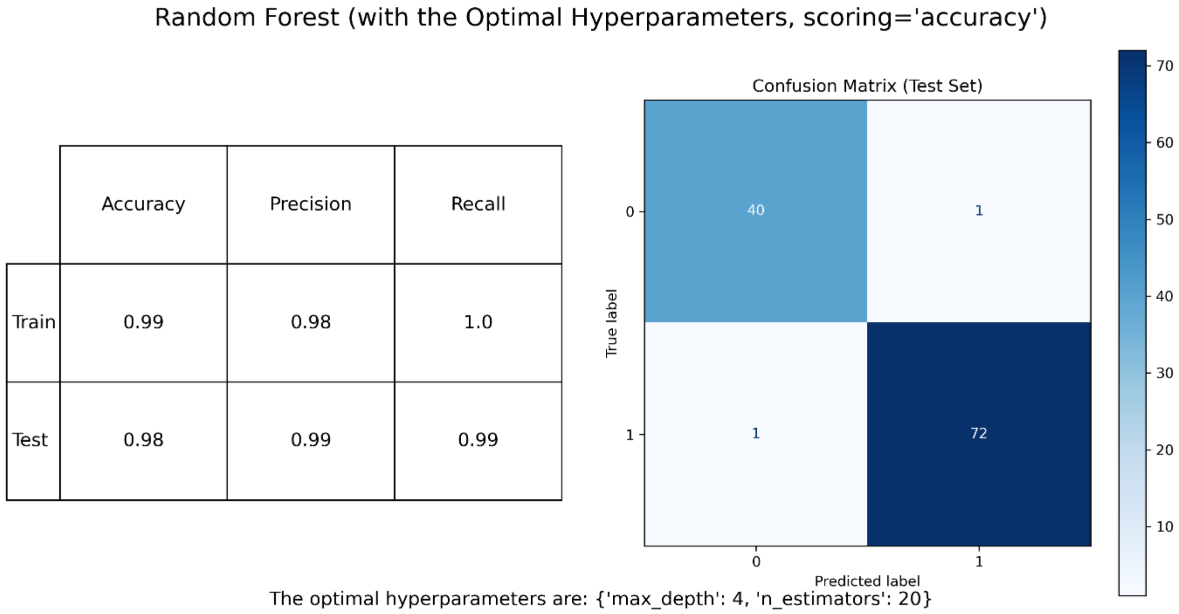
4.4 Task D18

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 enables comparison of using the full feature set and the reduced feature set.

Reducing the number of features does not have any impact on the model performance. According to D15, those reduced features all have a feature importance of 0, which means these features do not reduce impurity nor contribute to the prediction in each split.

5. Random Forest Models

5.1 Task D19



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

	Accuracy	Precision	Recall
Train	0.99	0.98	1.0
Test	0.98	0.99	0.99

The optimal hyperparameters are: {'max_depth': 4, 'n_estimators': 20}

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 performance obtained in D12. The FN and FP misclassifications are fewer than that of 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.

5.3 Task D21

I do not think these models are good enough and can be trusted to be used in real life. These models were trained based on a small set of data points. And through inspection in D1 we can observe anomalies and outliers existing in the dataset. These affects the prediction accuracy and its generalisation capability for unseen data.

A more complex model might not be necessary due to the high recall, accuracy, precision scores and good performance of confusion matrix, especially for the Random Forest model in D19.

I do not think using a machine learning algorithm for this task at this stage is a good idea. This decision process should not be automated. Identifying breast cancer is a crucial task and we do not want to misclassify any case. It is more suitable for being used to assist the manual diagnosis of breast cancer. We should not rely solely on these machine learning models.

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