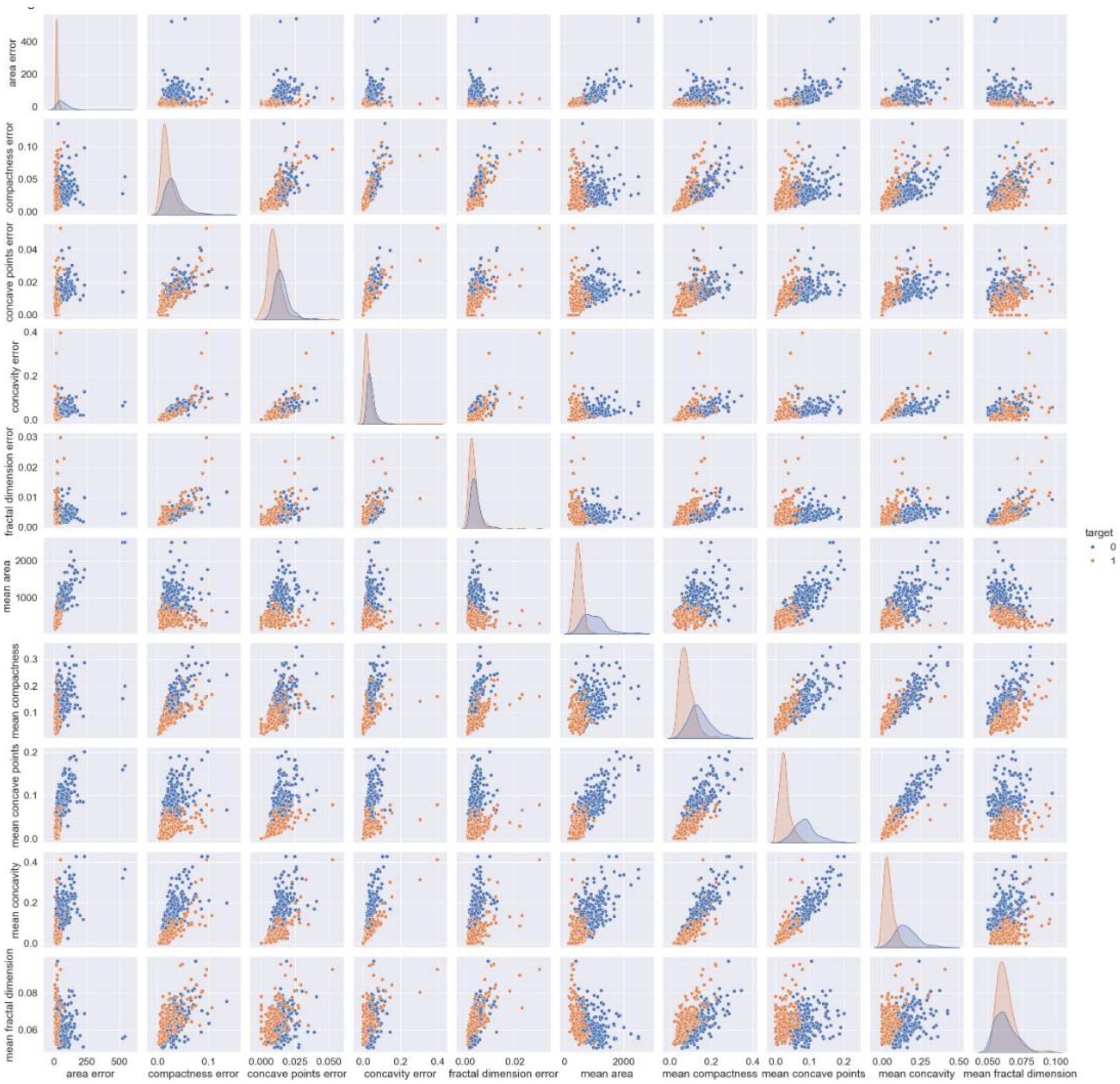
Zachary Tan (23177104)

Analysis

D1



D2

What can be observed regarding the relationship between these features?

- A lot of features have a linear relationship with other features. This is usually indicative of redundant features. For example, area and perimeter seem like they are linearly related.
- Some features have a non-linear relationship, showing unique information.

Can you observe the presence of clusters of groups? How do they relate to the target variable?

Zachary Tan (23177104)

- Some variables have clusters of groups of certain classes, showing that that variable differentiates well between the two classes. An example of which is mean area.

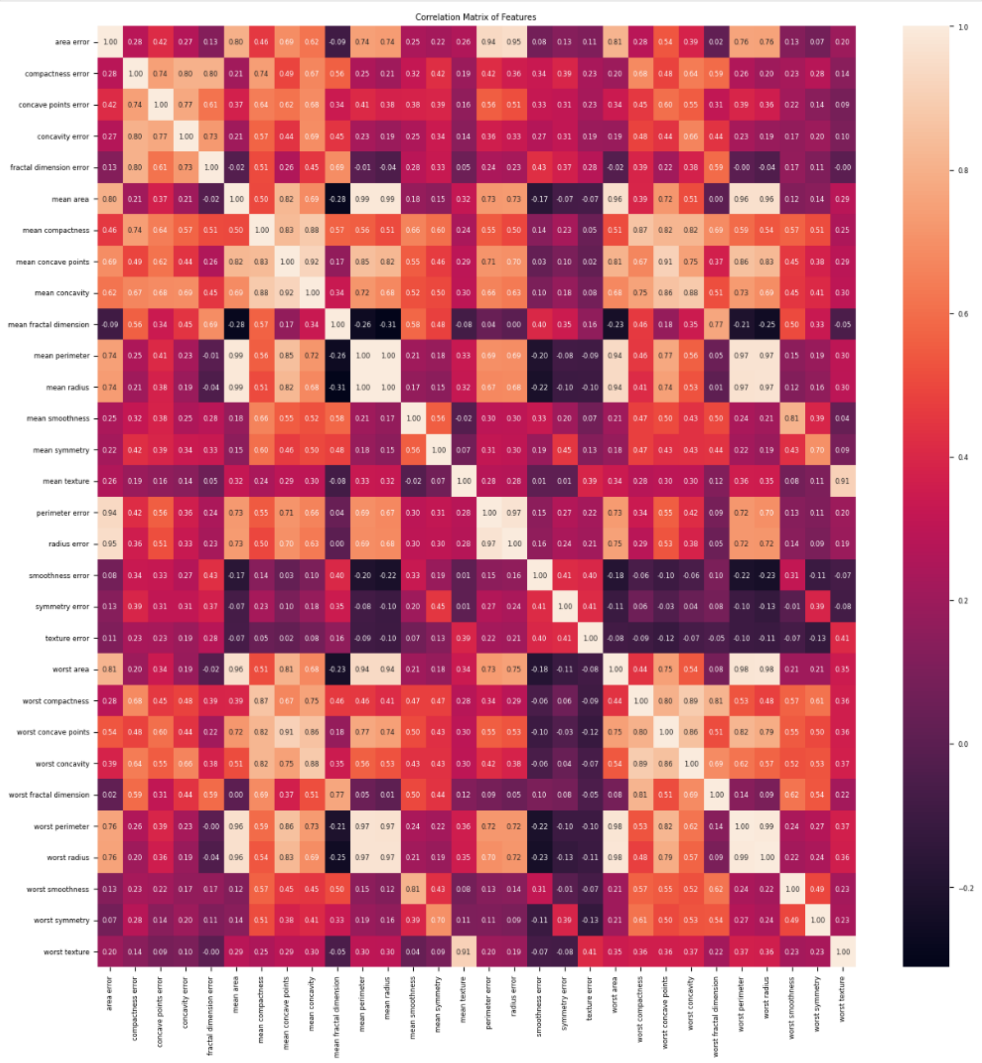
Are there any instances that could be outliers?

- There are heaps of instances that show way outside the normal grouping. An example of which is in mean concavity by mean fractal dimension.

Are there features that could be removed? Why or why not?

- Features which are highly correlated, then a feature can be removed to reduce complexity as they contain the same information.
- An example feature is mean fractal dimension or fractal dimension error, as they both appear to show similar information.
- Further information is required to be sure we can remove this data however, as this data may seem similar to the eye test but may prove to hold unique information still.

D3

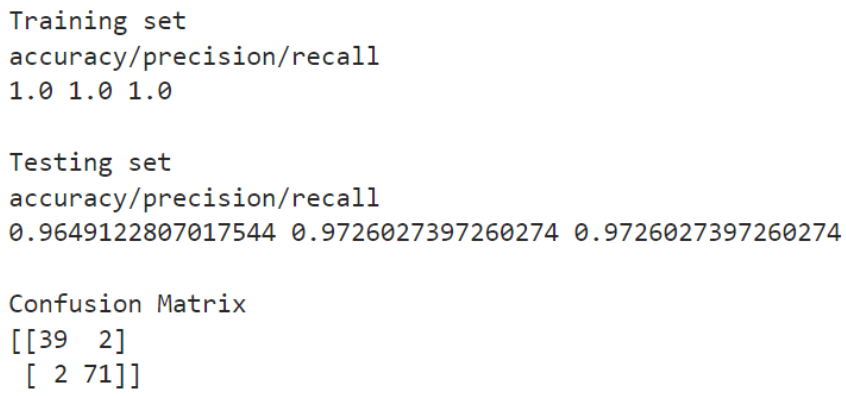


Zachary Tan (23177104)

D4

Whilst this did support my previous observations, with a correlation coefficient of 0.77, there were several relationships with higher correlation, such as mean radius, mean perimeter, worst perimeter, worst area and worst radius, all with correlations higher than 0.94, meaning that some of them are redundant, as they contain the same information

D6

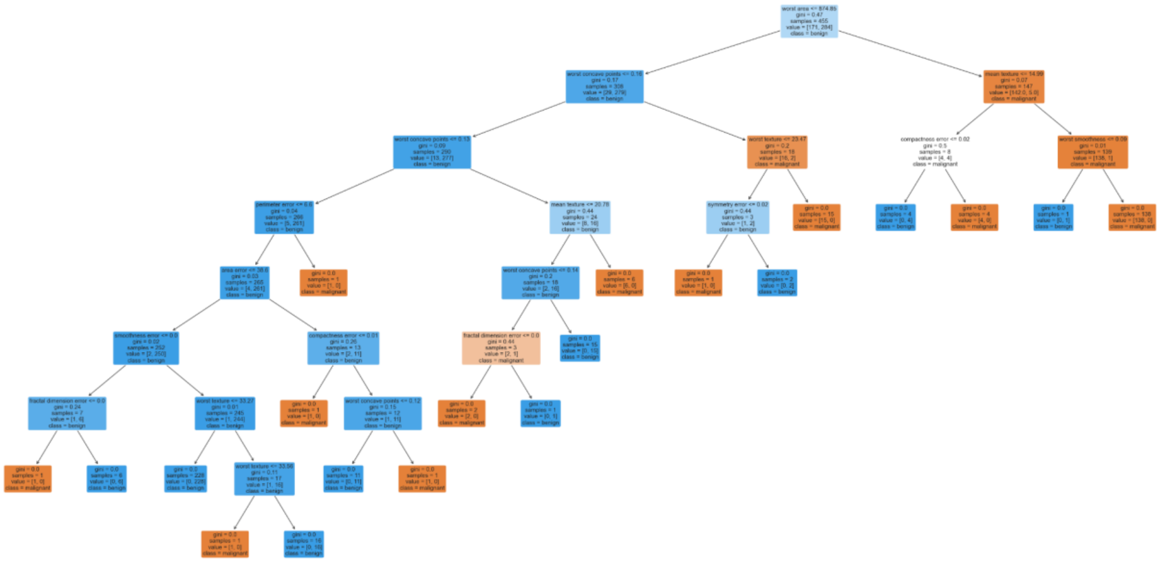


```
Training set
accuracy/precision/recall
1.0 1.0 1.0

Testing set
accuracy/precision/recall
0.9649122807017544 0.9726027397260274 0.9726027397260274

Confusion Matrix
[[39  2]
 [ 2 71]]
```

D7

The classifier is not overfitting the data as the accuracy, precision and recall of the Testing set (0.96, 0.97, 0.97) or unseen data is still very high, showing that the classifier is generalising the data well. From the confusion matrix, there are few false positives (2) and false negatives (2). Whilst this data fit perfectly with the training data, there isn't a large discrepancy between the training and testing set and definitely not enough to consider this overfitting.

D8

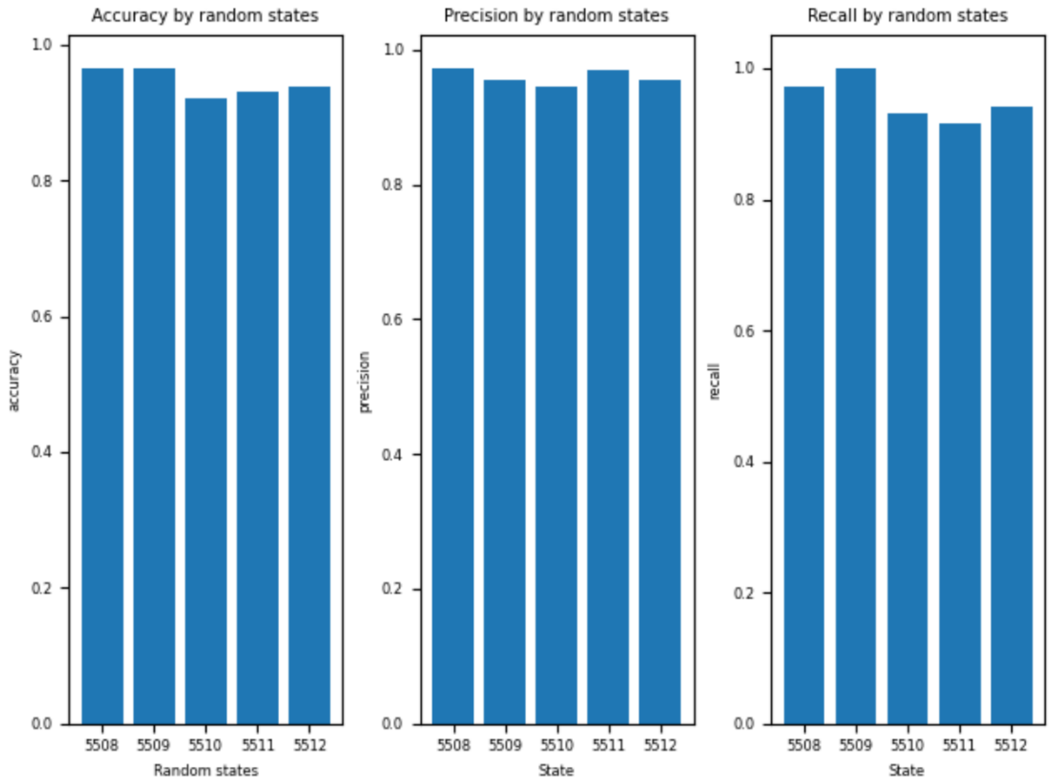

worst area <= 874.85 gini = 0.47 samples = 455 value = [171, 284] class = benign
worst concave points <= 0.16 gini = 0.17 samples = 308 value = [29, 279] class = benign
mean texture <= 14.99 gini = 0.07 samples = 147 value = [142.0, 5.0] class = malignant
worst concave points <= 0.13 gini = 0.09 samples = 290 value = [13, 277] class = benign
worst texture <= 23.47 gini = 0.2 samples = 18 value = [16, 2] class = malignant
compactness error <= 0.02 gini = 0.5 samples = 8 value = [4, 4] class = malignant
worst smoothness <= 0.09 gini = 0.01 samples = 139 value = [138, 1] class = malignant
perimeter error <= 6.6 gini = 0.04 samples = 266 value = [5, 261] class = benign
mean texture <= 20.78 gini = 0.44 samples = 24 value = [8, 16] class = benign
symmetry error <= 0.02 gini = 0.44 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 <= 38.6 gini = 0.03 samples = 265 value = [4, 261] class = benign
gini = 0.0 samples = 1 value = [1, 0] class = malignant
worst concave points <= 0.14 gini = 0.2 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 <= 0.0 gini = 0.02 samples = 252 value = [2, 250] class = benign
compactness error <= 0.01 gini = 0.26 samples = 13 value = [2, 11] class = benign
fractal dimension error <= 0.0 gini = 0.44 samples = 3 value = [2, 1] class = malignant
gini = 0.0 samples = 15 value = [0, 15] class = benign
fractal dimension error <= 0.0 gini = 0.24 samples = 7 value = [1, 6] class = benign
worst texture <= 33.27 gini = 0.01 samples = 245 value = [1, 244] class = benign
gini = 0.0 samples = 1 value = [1, 0] class = malignant
worst concave points <= 0.12 gini = 0.15 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 = 0.11 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

Zachary Tan (23177104)

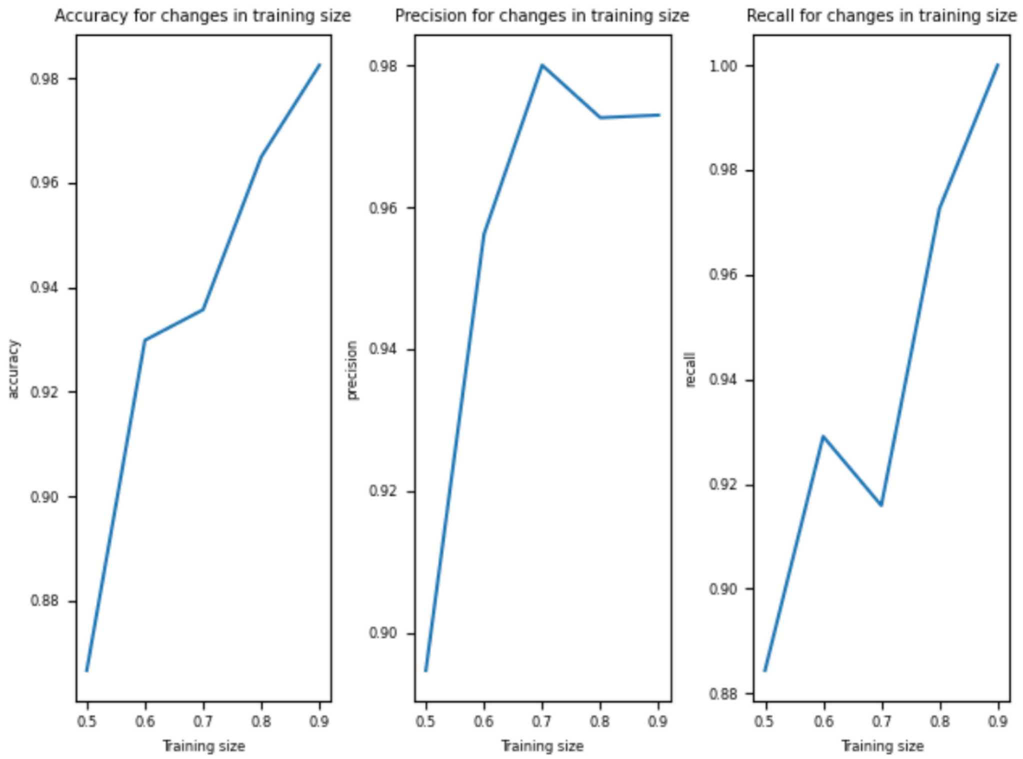
D9

There are 8 levels to this decision tree. This decision tree was quite complex, with 8 levels and many leaves. Complex decision trees (especially for small datasets) is typically considered as a sign for overfitting. As for the leaves, whilst there are a few leaves with a small sample size (typical of overfitting), most leaves have a decent sample size, lessening the sign for overfitting. This decision tree is somewhat interpretable, as it is quite complex.

D10

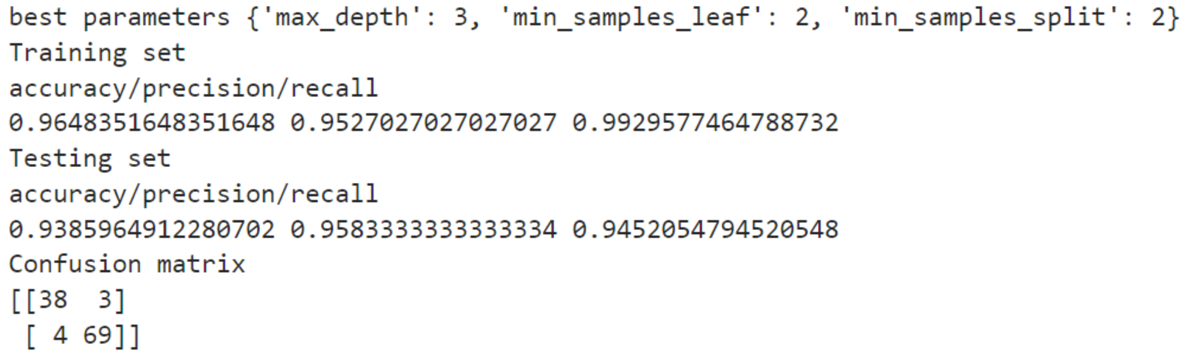


D11



It behaved as expected – with increasing training size, the accuracy precision and recall of the testing set increase to a certain extent. Specifically with precision, we can see that it plateaued around 70%-80% training size, possibly indicating overfitting of the model once the training size has reached its optimal stage.

D12



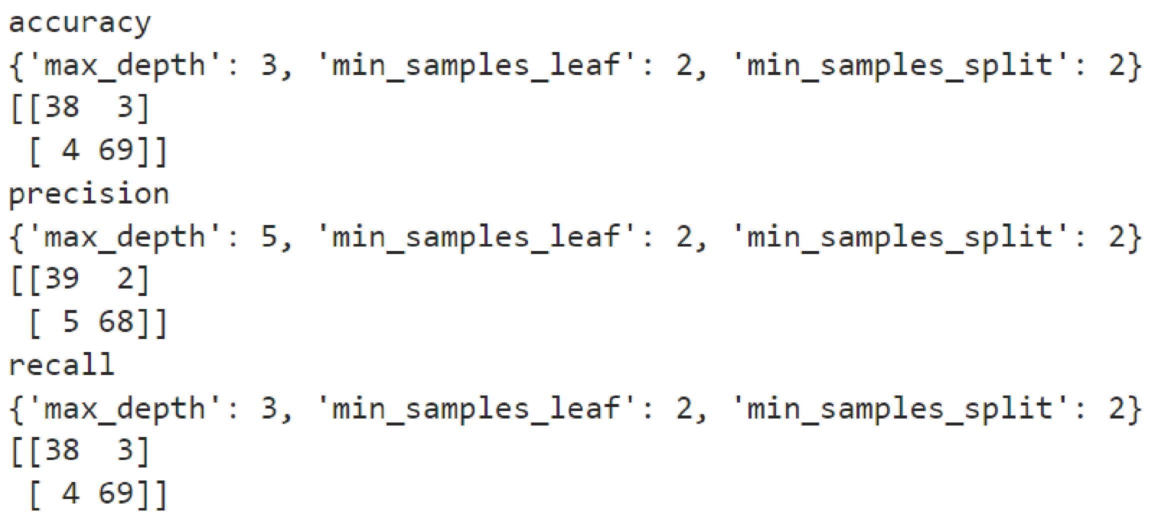
```
best parameters {'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}
Training set
accuracy/precision/recall
0.9648351648351648 0.9527027027027027 0.9929577464788732
Testing set
accuracy/precision/recall
0.9385964912280702 0.9583333333333334 0.9452054794520548
Confusion matrix
[[38  3]
 [ 4 69]]
```

D13

We can see that accuracy, precision and recall decreased across the training and testing set with more false positives and false negatives by applying these optimal hyperparameters. In other words, there is a decrease in performance. In this case, fine-tuning has not done what was expected.

Zachary Tan (23177104)

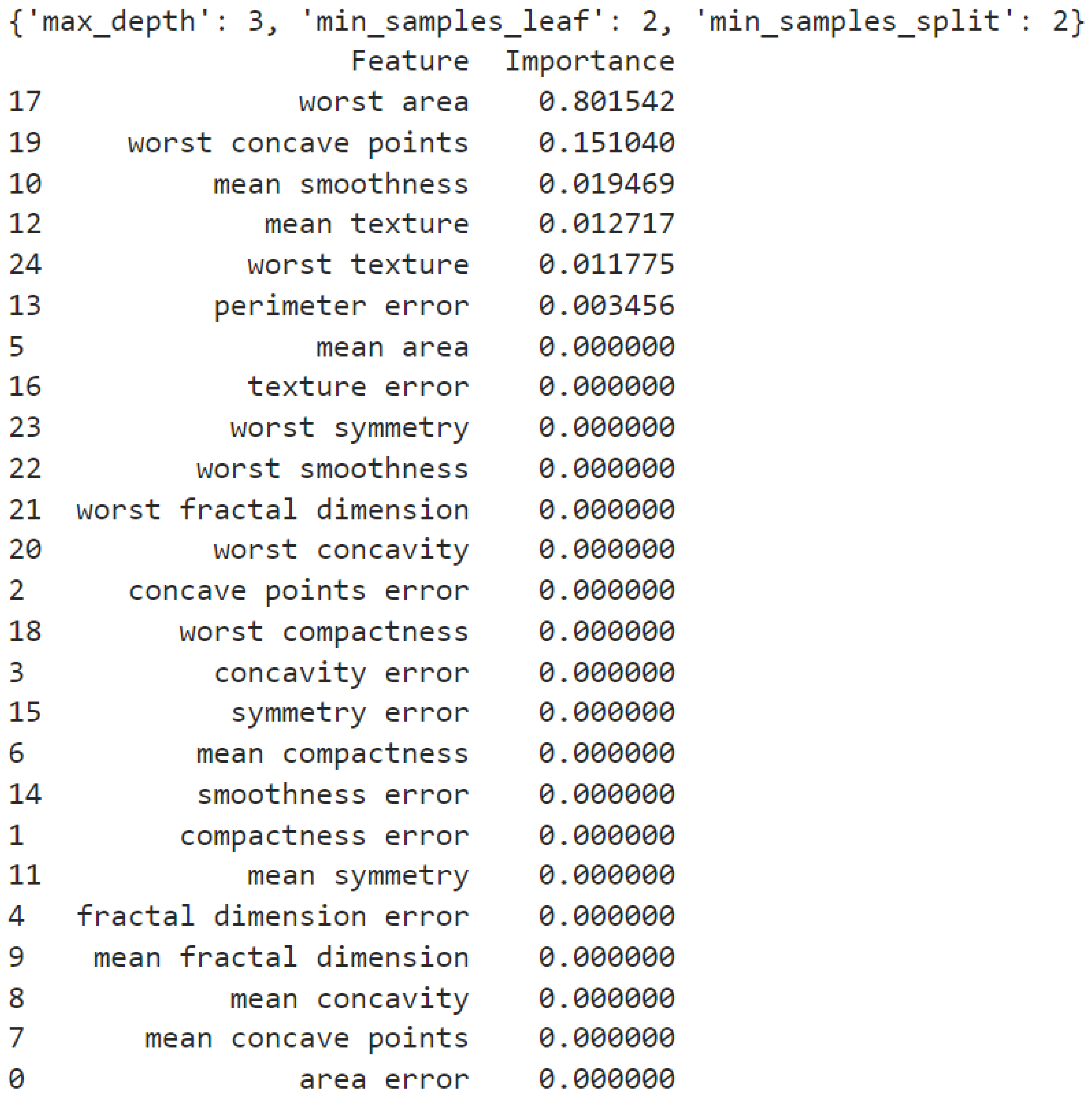
D14



```
accuracy
{'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}
[[38  3]
 [ 4 69]]
precision
{'max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 2}
[[39  2]
 [ 5 68]]
recall
{'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}
[[38  3]
 [ 4 69]]
```

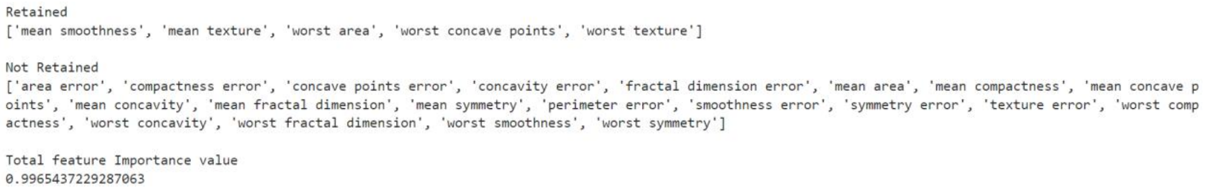
If we consider the problem as a whole, rather than specifically for accuracy, precision or recall, a deeper tree seems to make the classifier more precise. Optimising specifically for any of these metrics did not seem to make a noticeable improvement in the classifier over another.

D15



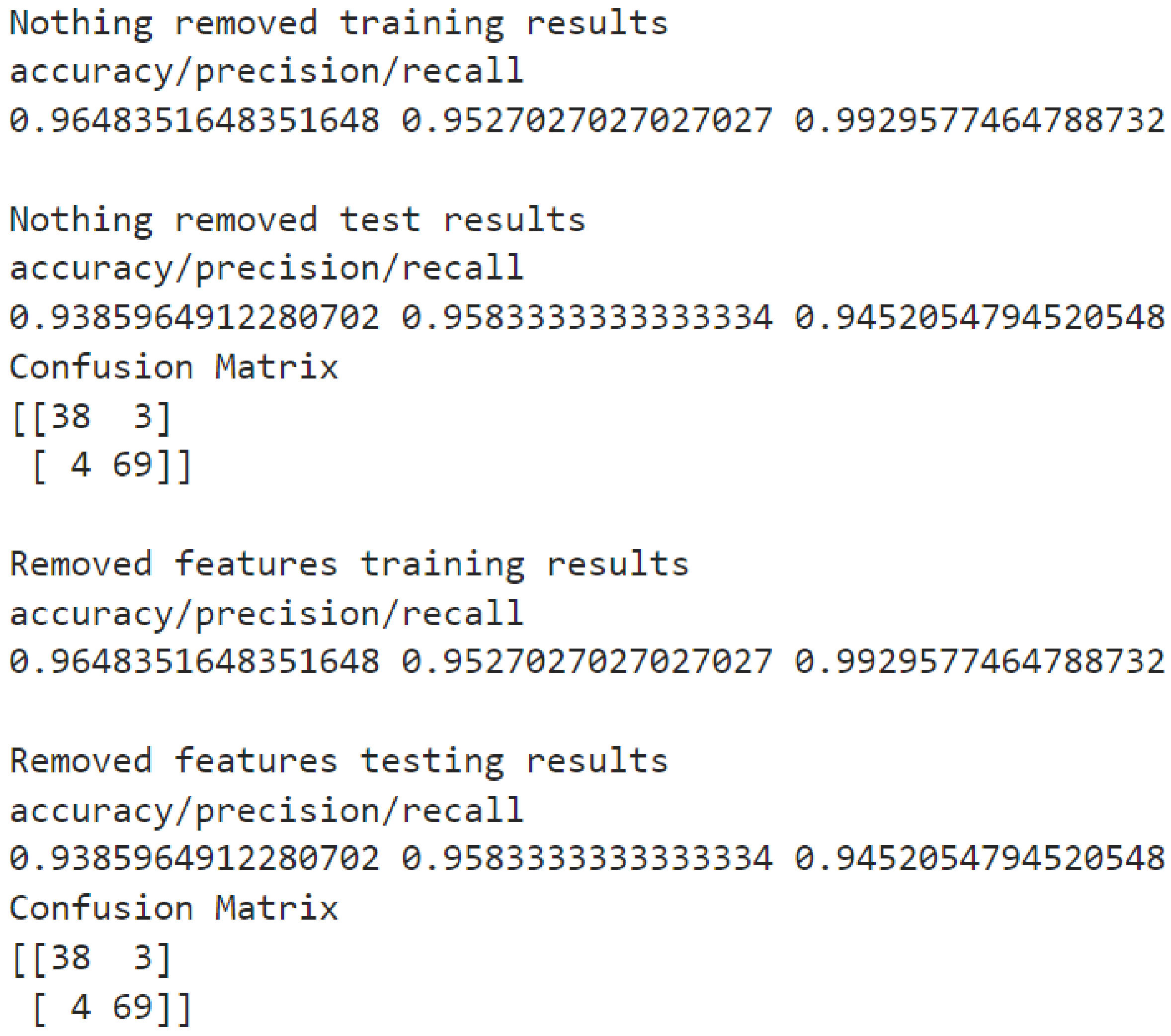
```
{'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 2}
                   Feature  Importance
17              worst area    0.801542
19    worst concave points    0.151040
10         mean smoothness    0.019469
12            mean texture    0.012717
24           worst texture    0.011775
13         perimeter error    0.003456
5                mean area    0.000000
16           texture error    0.000000
23          worst symmetry    0.000000
22        worst smoothness    0.000000
21 worst fractal dimension    0.000000
20         worst concavity    0.000000
2     concave points error    0.000000
18       worst compactness    0.000000
3          concavity error    0.000000
15          symmetry error    0.000000
6         mean compactness    0.000000
14        smoothness error    0.000000
1        compactness error    0.000000
11           mean symmetry    0.000000
4  fractal dimension error    0.000000
9   mean fractal dimension    0.000000
8           mean concavity    0.000000
7      mean concave points    0.000000
0               area error    0.000000
```

D16



```
Retained
['mean smoothness', 'mean texture', 'worst area', 'worst concave points', 'worst texture']

Not Retained
['area error', 'compactness error', 'concave points error', 'concavity error', 'fractal dimension error', 'mean area', 'mean compactness', 'mean concave p
oints', 'mean concavity', 'mean fractal dimension', 'mean symmetry', 'perimeter error', 'smoothness error', 'symmetry error', 'texture error', 'worst comp
actness', 'worst concavity', 'worst fractal dimension', 'worst smoothness', 'worst symmetry']

Total feature Importance value
0.9965437229287063
```

D17

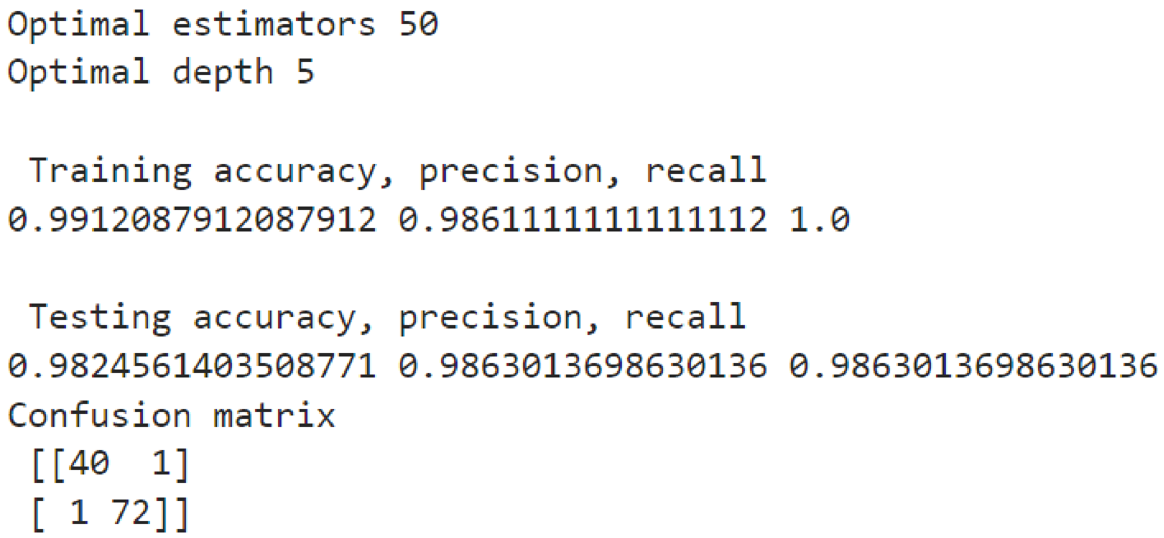


```
Nothing removed training results
accuracy/precision/recall
0.9648351648351648 0.9527027027027027 0.9929577464788732

Nothing removed test results
accuracy/precision/recall
0.9385964912280702 0.9583333333333334 0.9452054794520548
Confusion Matrix
[[38  3]
 [ 4 69]]

Removed features training results
accuracy/precision/recall
0.9648351648351648 0.9527027027027027 0.9929577464788732

Removed features testing results
accuracy/precision/recall
0.9385964912280702 0.9583333333333334 0.9452054794520548
Confusion Matrix
[[38  3]
 [ 4 69]]
```

D18

Considering the total feature importance sum of 0.9965 shows that we lost basically no information that was useful to the classifier by removing these features. Similarly, the accuracy, precision and recall were all the same with the reduced features, showing that there was no impact in removing these features.

D19



```
Optimal estimators 50
Optimal depth 5

 Training accuracy, precision, recall
0.9912087912087912 0.9861111111111112 1.0

 Testing accuracy, precision, recall
0.9824561403508771 0.9863013698630136 0.9863013698630136
Confusion matrix
 [[40  1]
 [ 1 72]]
```

D20

There was a reduction in the number of false positives and false negatives, as well as a higher accuracy, precision and recall when utilising the random forest model. This is expected behaviour as the random forest model generally performs better by building multiple trees, giving a more normalised result. This reduces the impact of outliers.

D21

Some models, particularly the random forest model show good results and can be used practically. We should however consider the use case of this model in that it is to be used in a medical application. This means that we should emphasise high recall such that the rate of false negatives is low, as missing a diagnosis would have highly negative impacts. We should consider a more complex model such that we have a higher recall rate. I think a machine learning algorithm is a good idea in this context, although not to replace the role of a medical practitioner. Rather, as a tool to help diagnose. Solely relying on an algorithm to make a diagnosis is unethical as patients should be confident that they are receiving the best possible diagnosis possible. As such doctors should be prioritised in diagnosis'.

The used data set is fine and well documented. We should ensure that this data set is a representative set of the actual population if using models based on this data set.