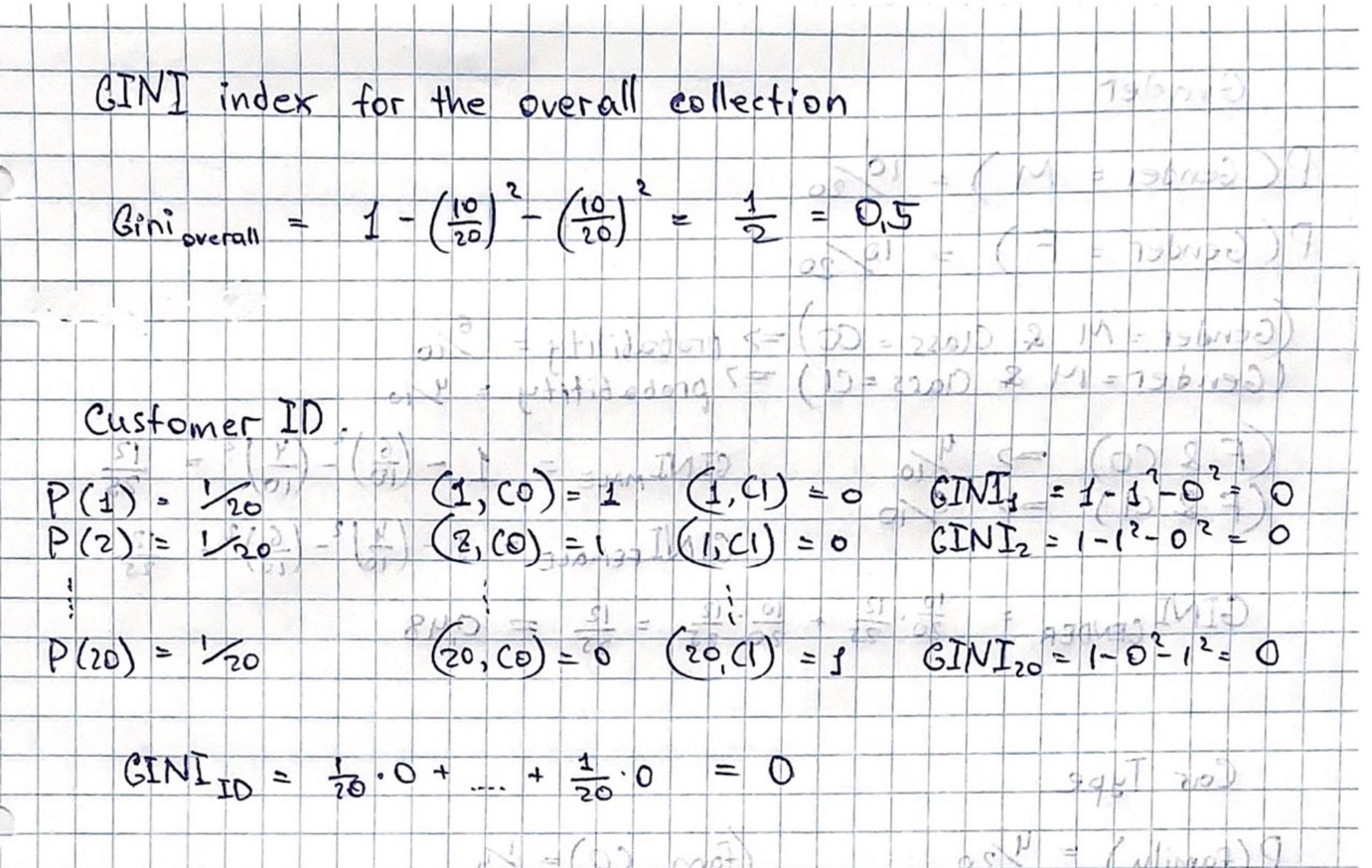
Assignment – Decision Trees

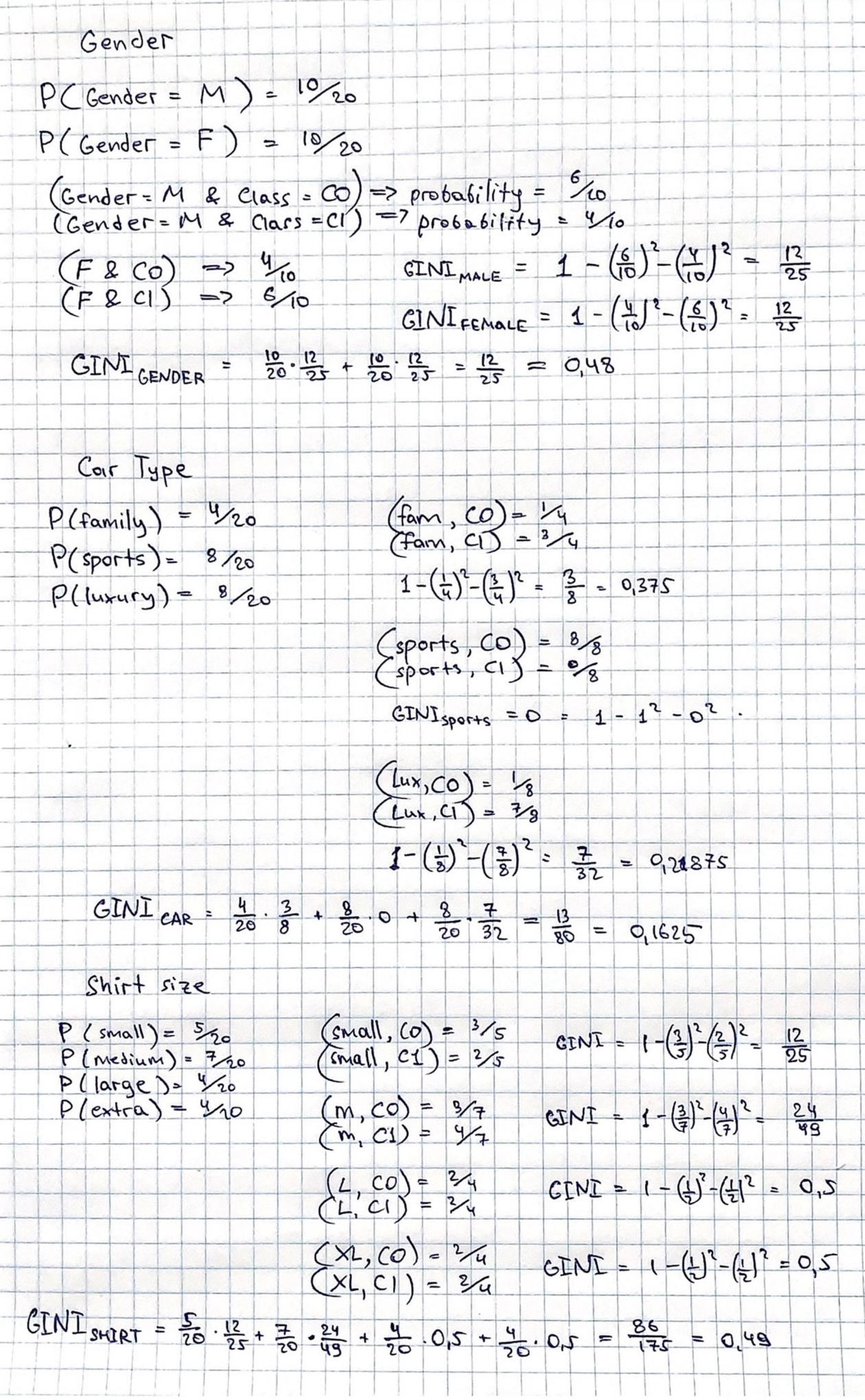
Rogov Sergei

Activity 1

1. b)



1. d) e)



f) Car Type is the best attribute since it has the lowest Gini index (0.1625) from these three attributes. [Gender is 0.48 and Shirt Size is 0.49]

g) Every customer is assigned to a unique ID number, that is why this attribute should not be considered. Customer ID attribute is not predictive.

Activity 2

Wine dataset (3 classes)

Report on accuracy with default parameters:

Accuracy (correct/total) = 0.9074074074074074

Scikit accuracy = 0.9074074074074074

Confusion Matrix

[[18, 1]

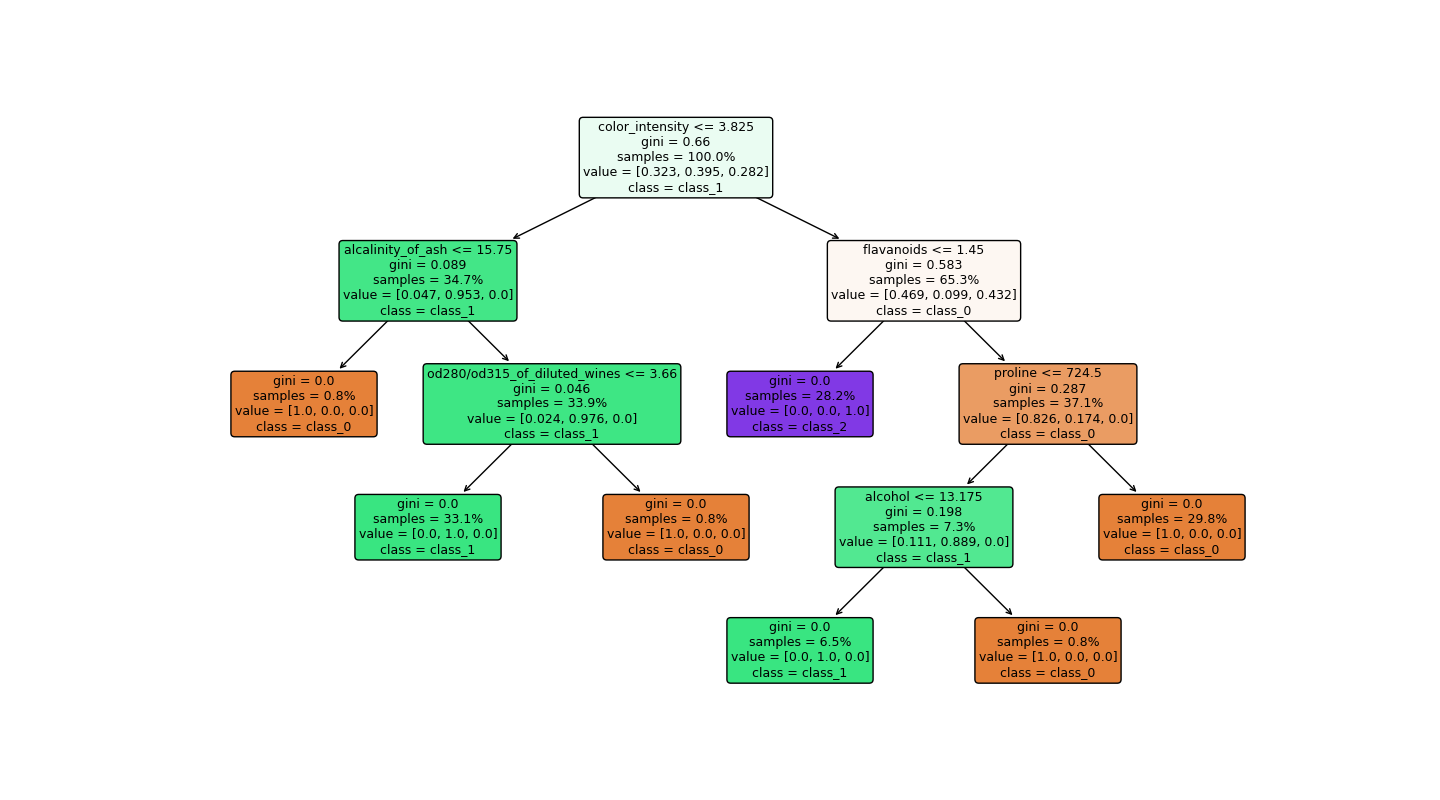
[ 2, 19]]

precision recall f1-score

0 0.86 0.95 0.90

1 0.95 0.86 0.90

2 0.92 0.92 0.92

Decision tree itself: 

IMPORTANCE OF ATTRIBUTES:

flavanoids 0.415881

color\_intensity 0.376051

proline 0.139755

od280/od315\_of\_diluted\_wines 0.023852

alcalinity\_of\_ash 0.022742

alcohol 0.021719

malic\_acid 0.000000

ash 0.000000

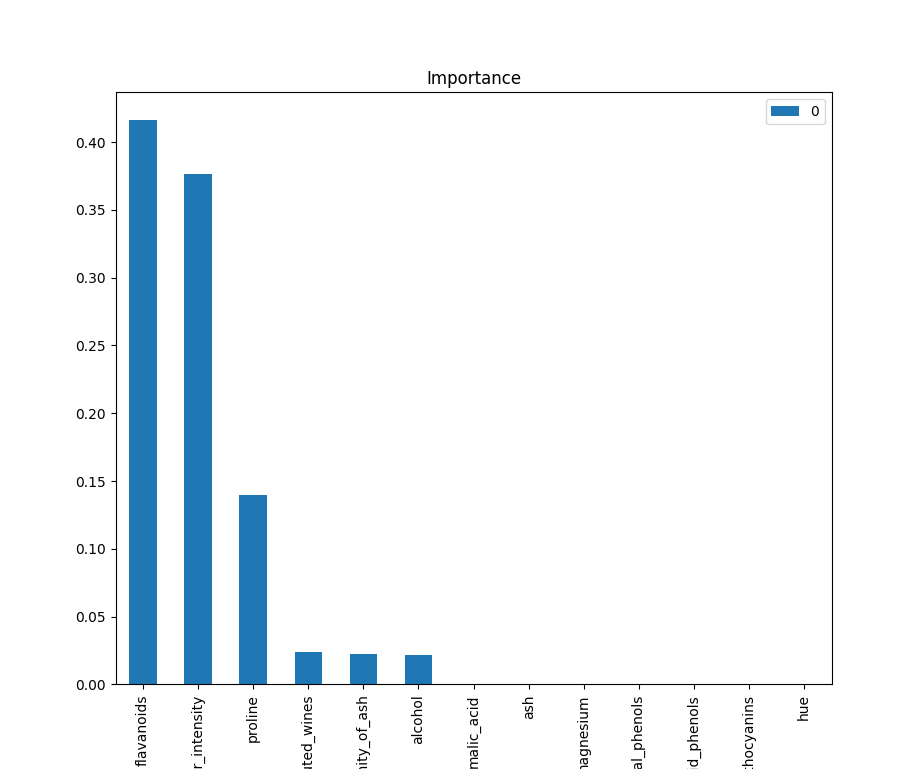
magnesium 0.000000

total\_phenols 0.000000

nonflavanoid\_phenols 0.000000

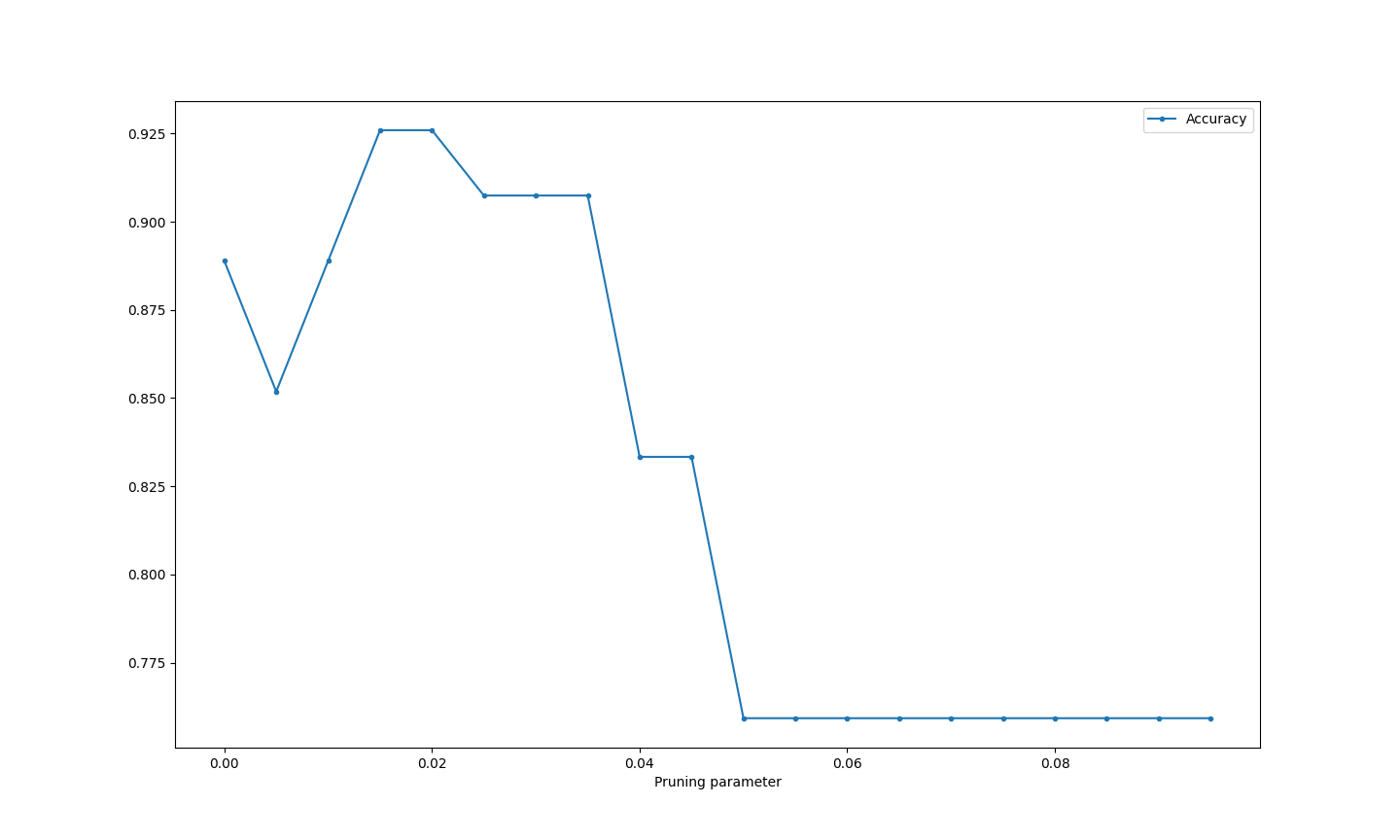
proanthocyanins 0.000000

hue 0.000000



Experimenting with parameters:

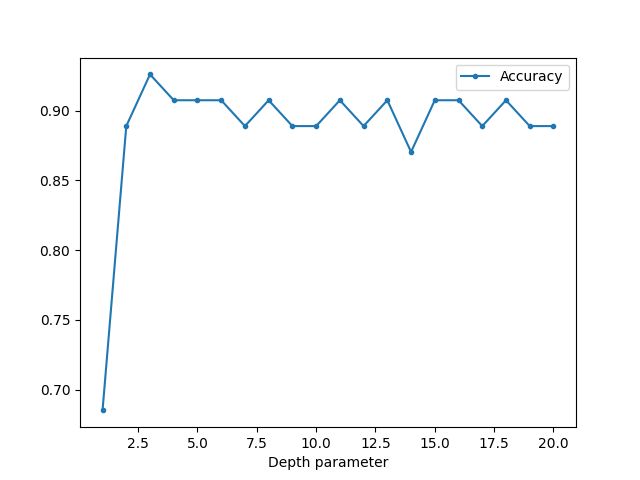
1. I varied ccp\_alpha parameter which is responsible for pruning the tree from 0 to 0.1 with a step of 0.005. The resulting plot looks like this:



Interpretation of the result: with a little amount of pruning the accuracy might be increased, but if you prune too much accuracy significantly decreases – we are losing a lot of information.

The reason why the accuracy at ccp\_alpha=0 does not equal 0.907 (as with default parameters) is because for splitting dataset into train and split sets I used a built-in train\_test\_split Scikit Learn function. It forms these groups differently every other time, that’s why the model is trained differently and is not exactly the same every time.

1. I varied max\_depth parameter which determines the maximum depth of the tree from 1 to 20 with a step of 1. The resulting plot:



Interpretation: if a tree has only 1 level, the accuracy is low, but it increases if a tree has a bigger depth.

However, too many levels might cause overfitting. The reason why it does not really happen here is because even with default parameters our tree only reaches 5 levels (the dataset is small).

Breast cancer dataset (2 classes, larger number of attributes and instances)

Report on accuracy with default parameters:

Accuracy = 0.9473684210526315

Scikit accuracy = 0.9473684210526315

Confusion Matrix

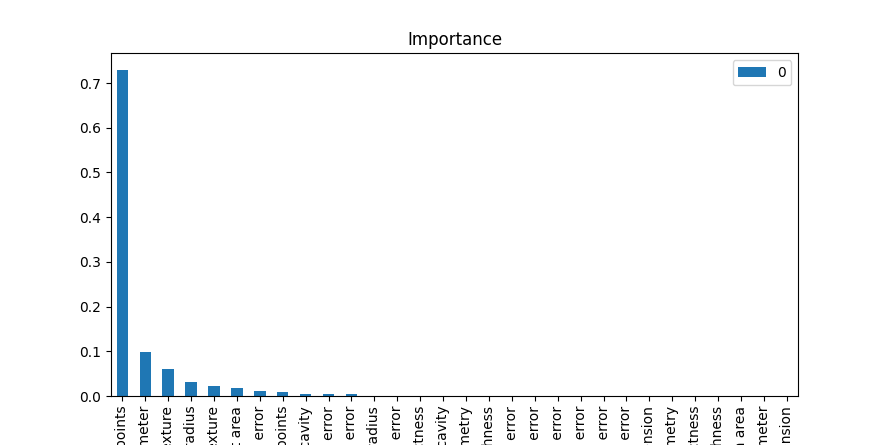
[[ 58 4]

[ 5 104]]

precision recall f1-score

0 0.92 0.94 0.93

1 0.96 0.95 0.96



IMPORTANCE OF ATTRIBUTES:

mean concave points 0.729749

worst perimeter 0.099591

worst texture 0.061548

worst radius 0.032303

mean texture 0.022109

worst area 0.018341

fractal dimension error 0.010467

worst concave points 0.009986

mean concavity 0.005349

perimeter error 0.005349

area error 0.005207

mean radius 0.000000

symmetry error 0.000000

worst compactness 0.000000

worst concavity 0.000000

worst symmetry 0.000000

worst smoothness 0.000000

compactness error 0.000000

concave points error 0.000000

concavity error 0.000000

smoothness error 0.000000

texture error 0.000000

radius error 0.000000

mean fractal dimension 0.000000

mean symmetry 0.000000

mean compactness 0.000000

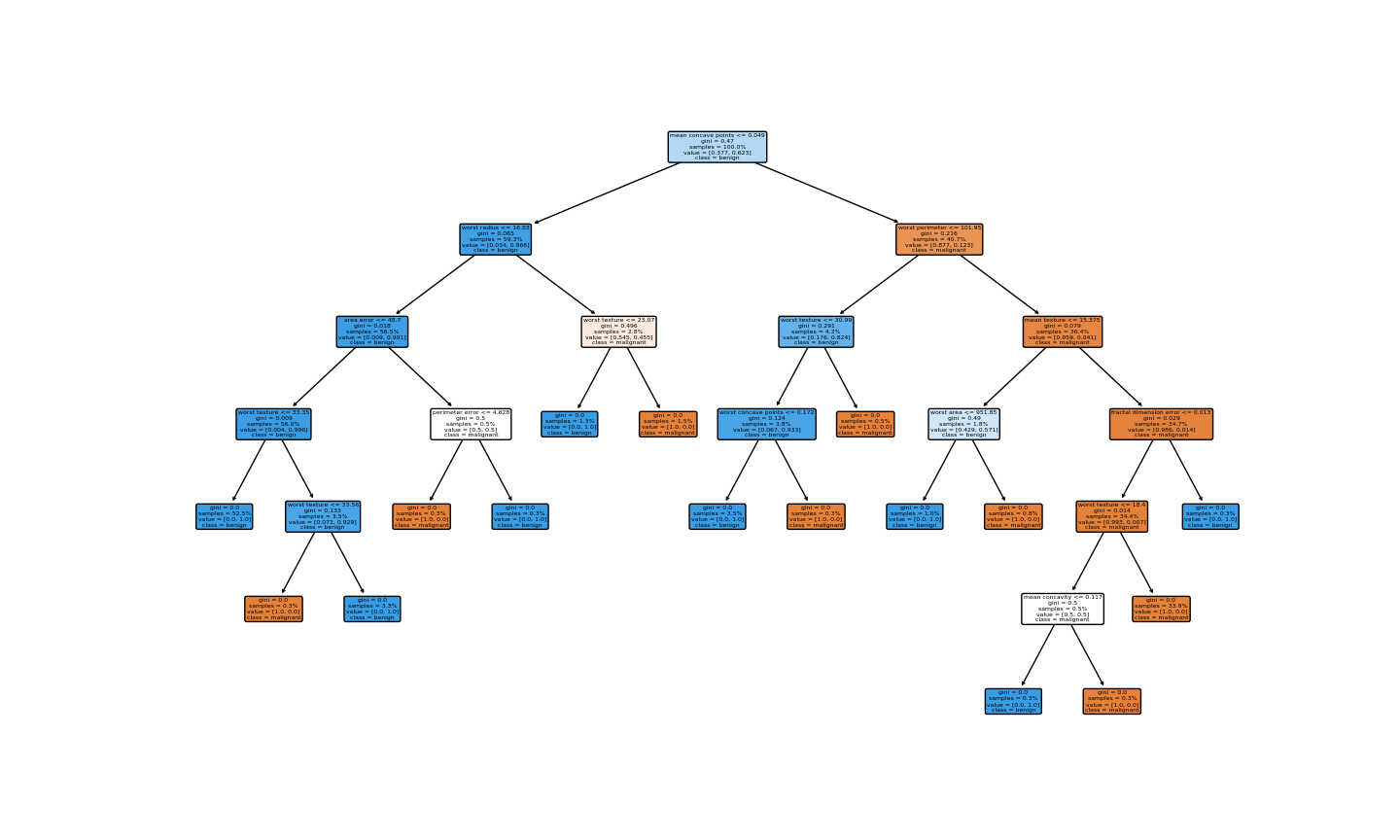
mean smoothness 0.000000

mean area 0.000000

mean perimeter 0.000000

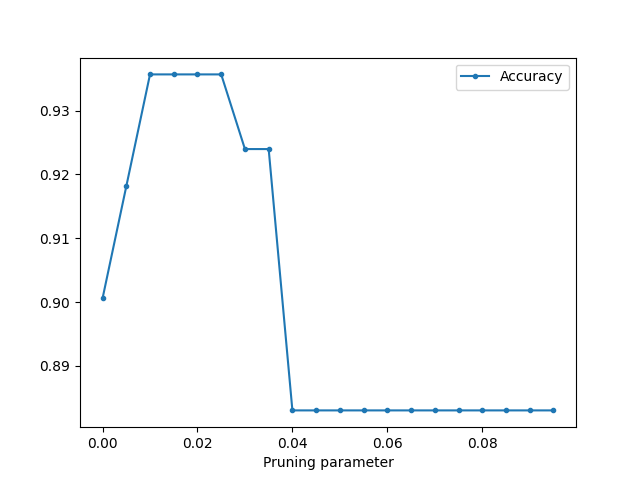
worst fractal dimension 0.000000

Decision tree:



Experimenting with parameters:

How accuracy depend on pruning parameter:



How accuracy depend on maximum depth parameter:

