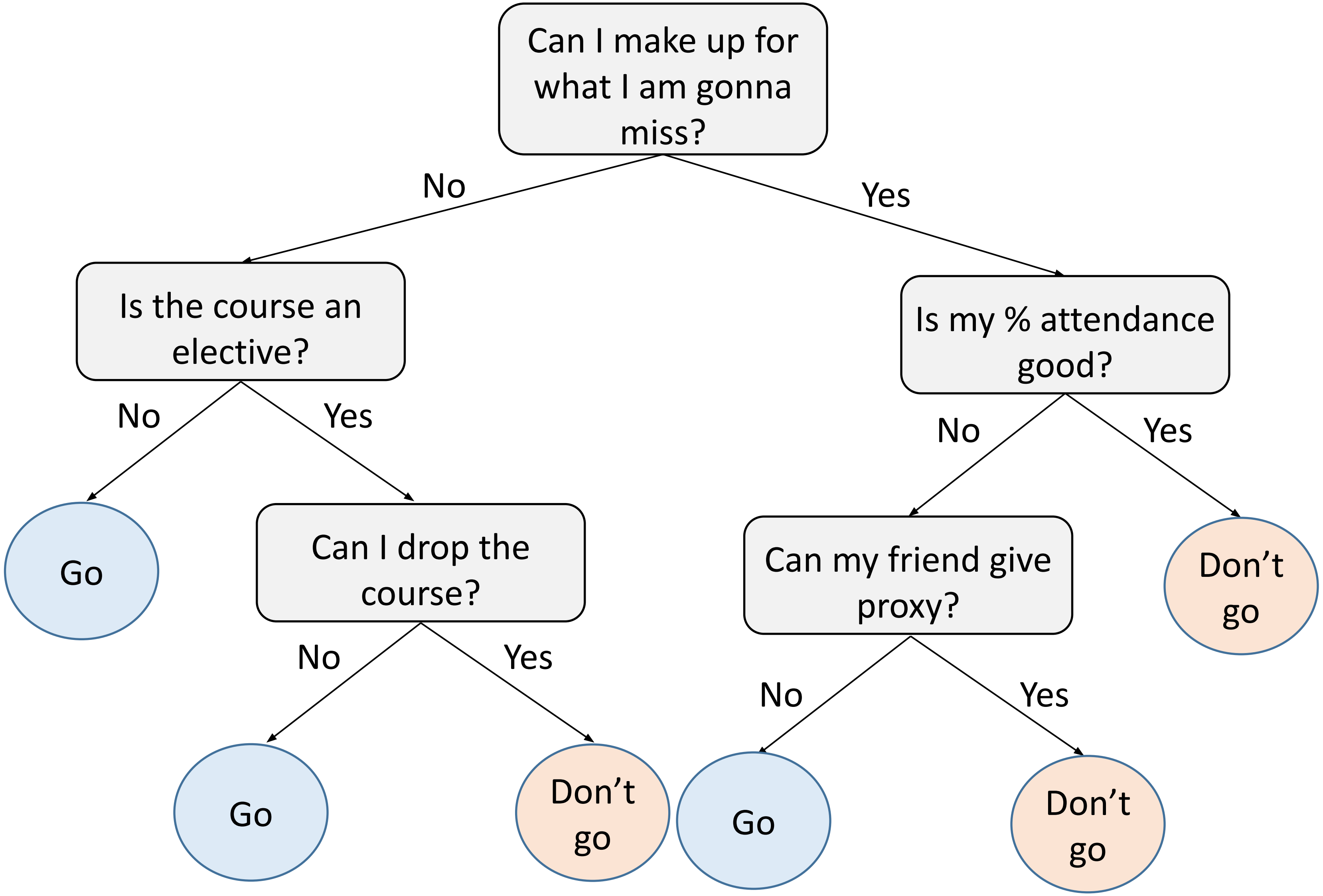
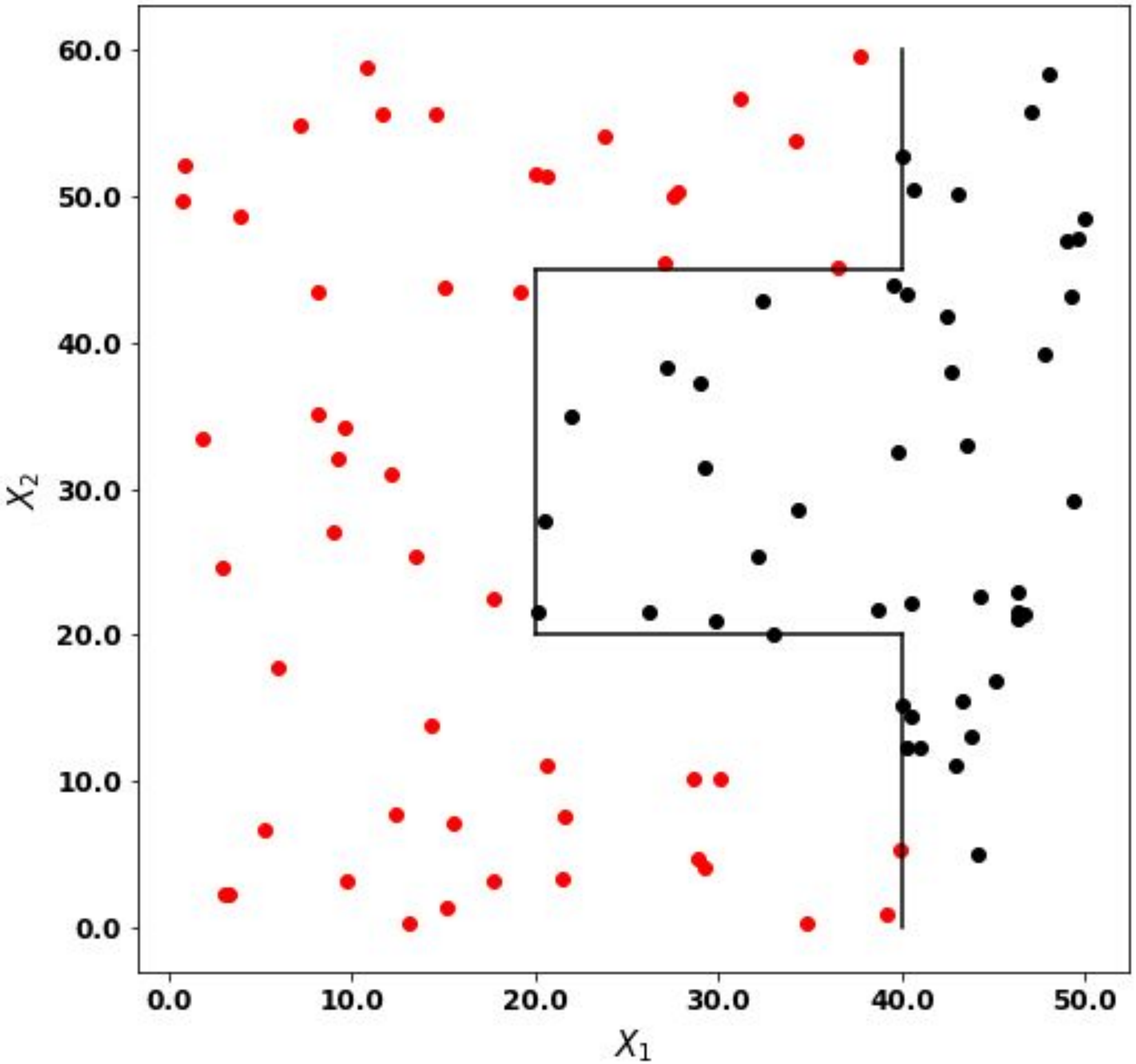
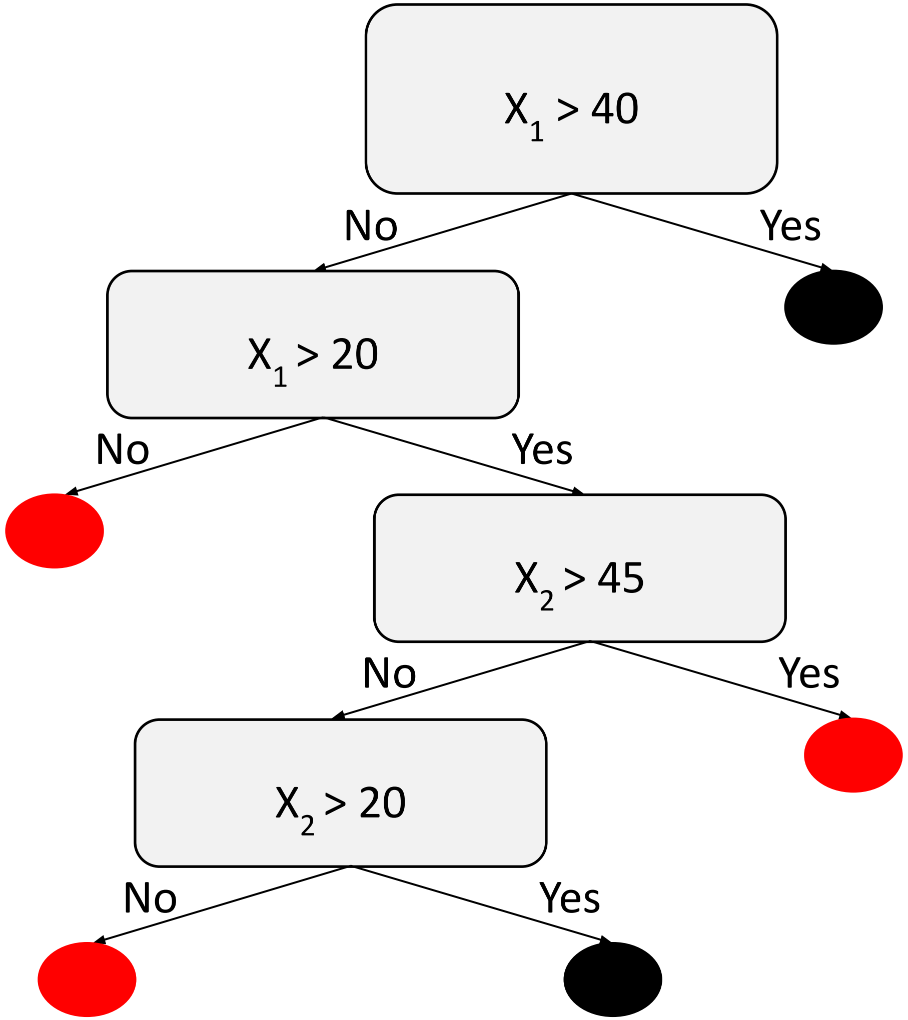
Should I need to go to class today?



Red or black?

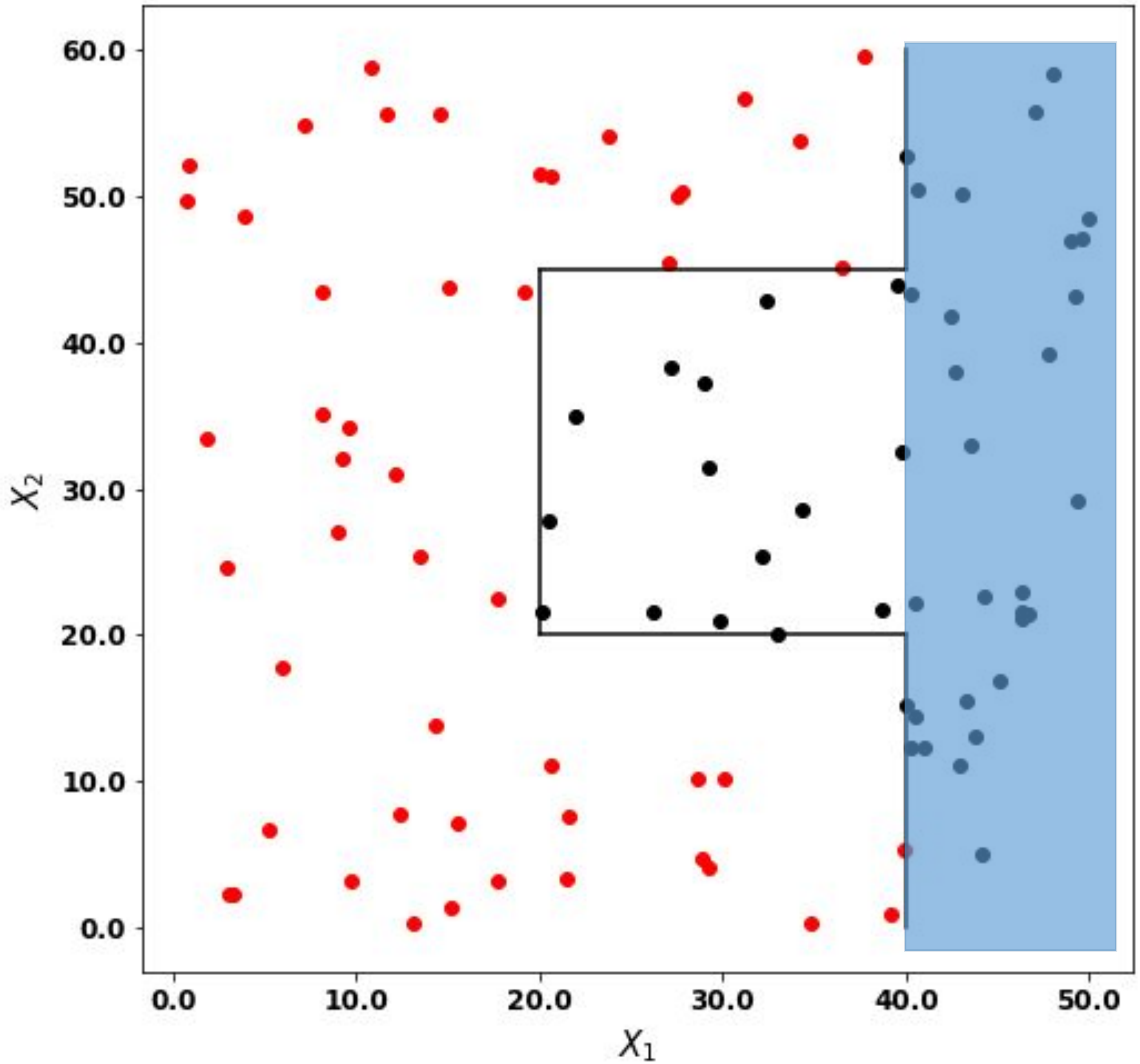


60.0
50.0
40.0
30.0
20.0
10.0
0.0
X_2
0.0
10.0
20.0
30.0
40.0
50.0
X_1

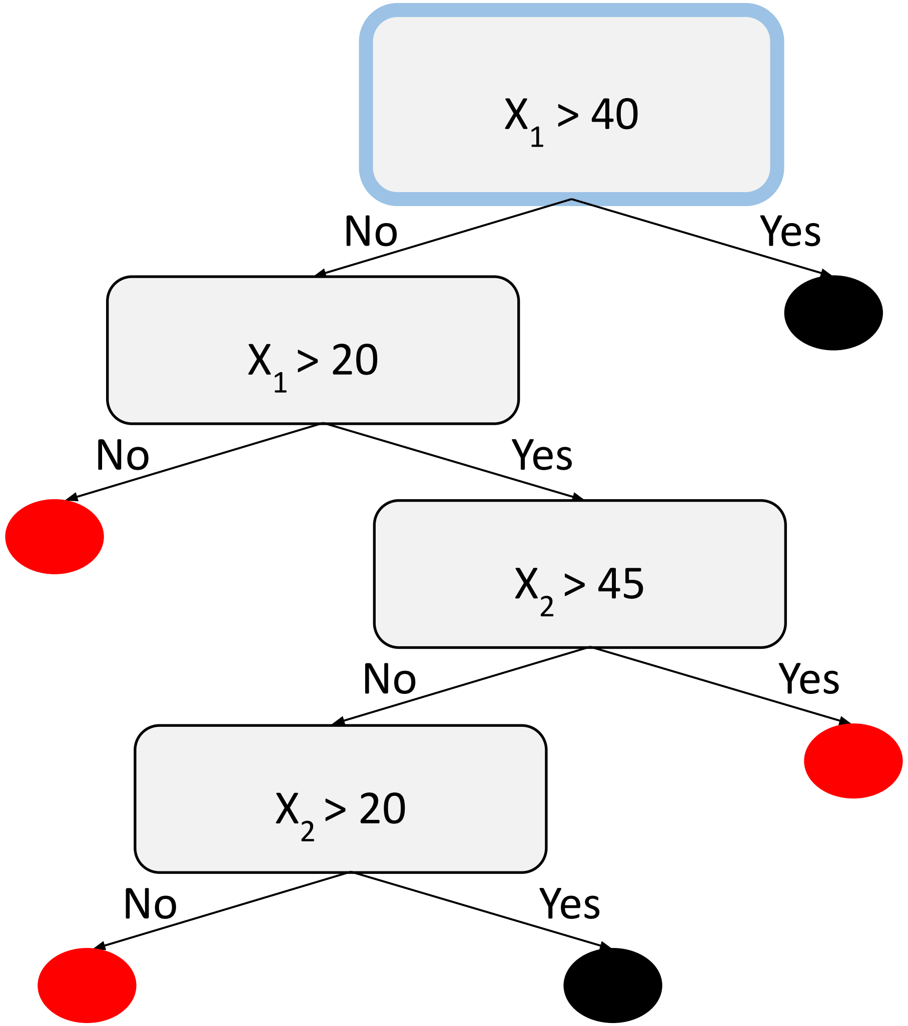


$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes
$X_2 > 45$
No
Yes
$X_2 > 20$
No
Yes

Red or black?

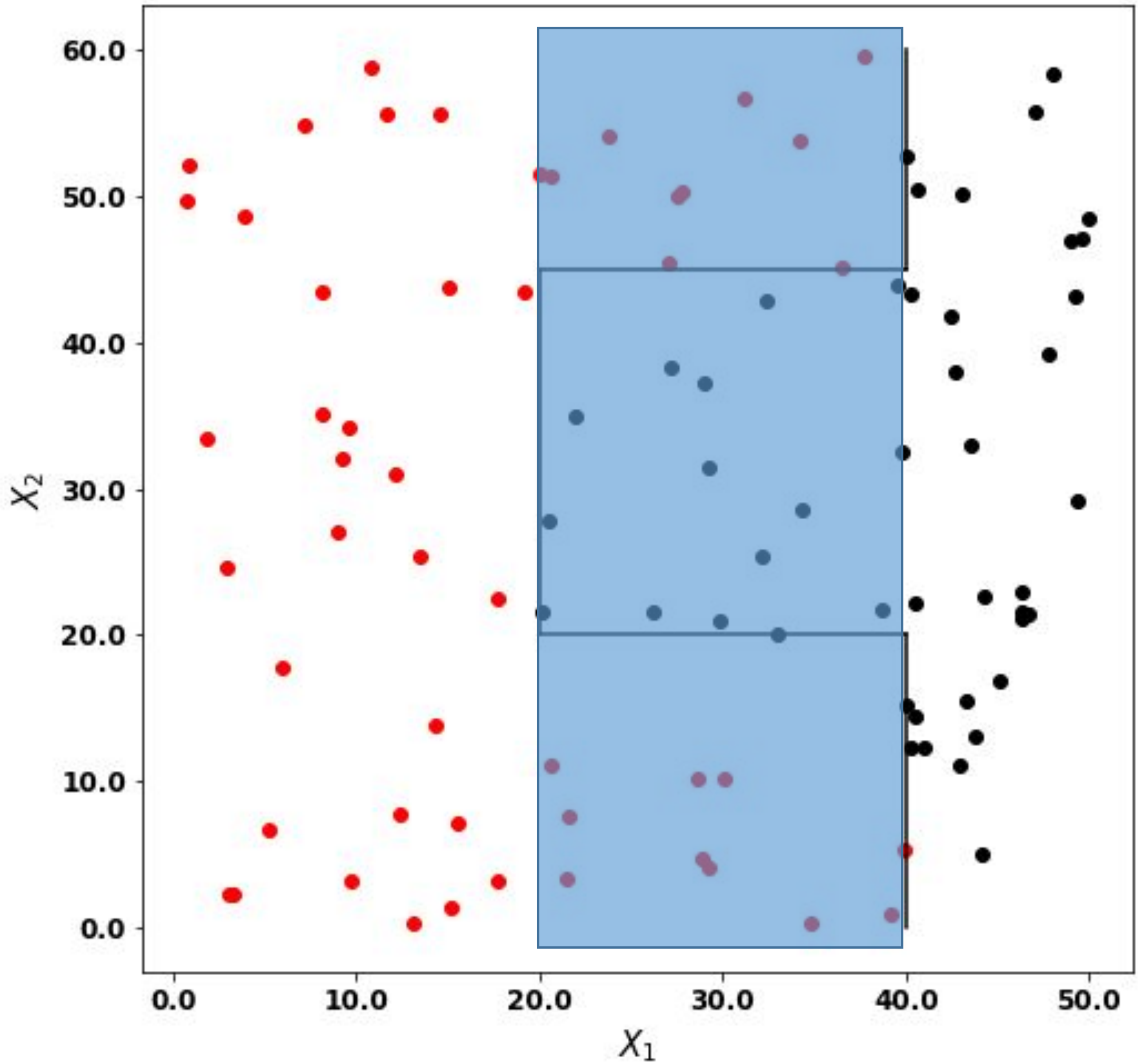


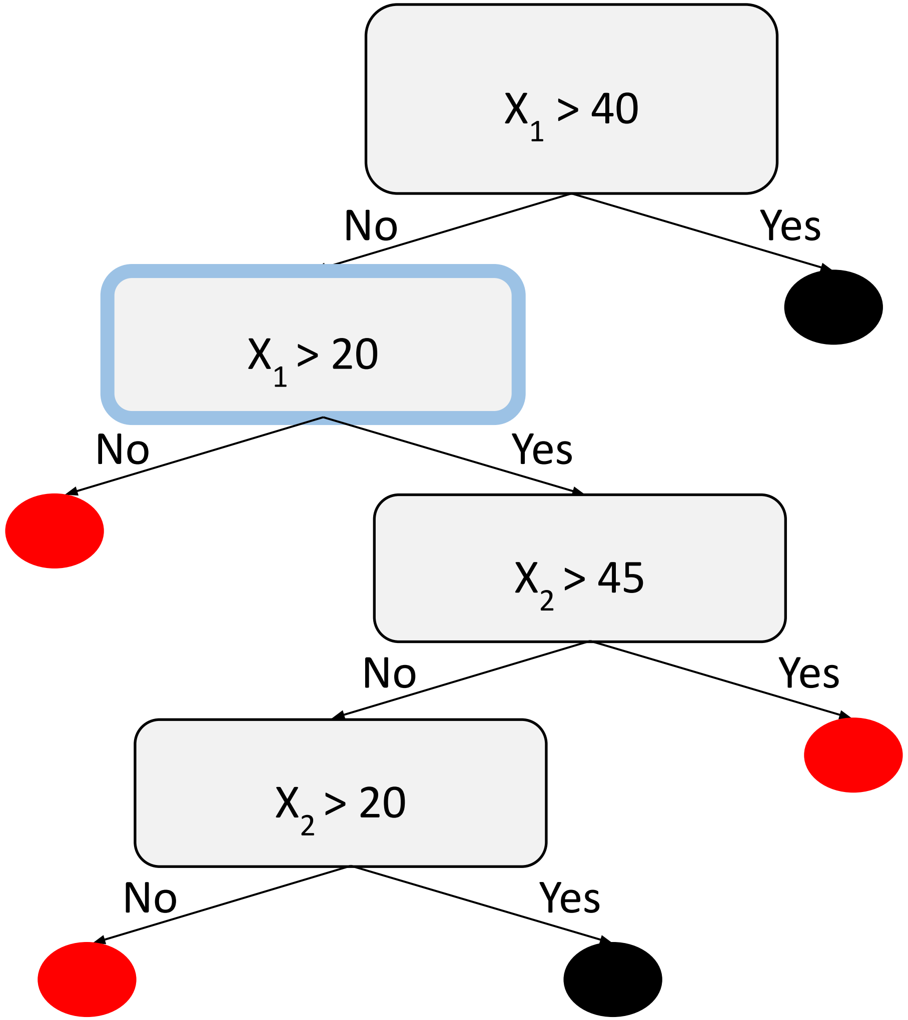
60.0
50.0
40.0
30.0
20.0
10.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
X_1
X_2



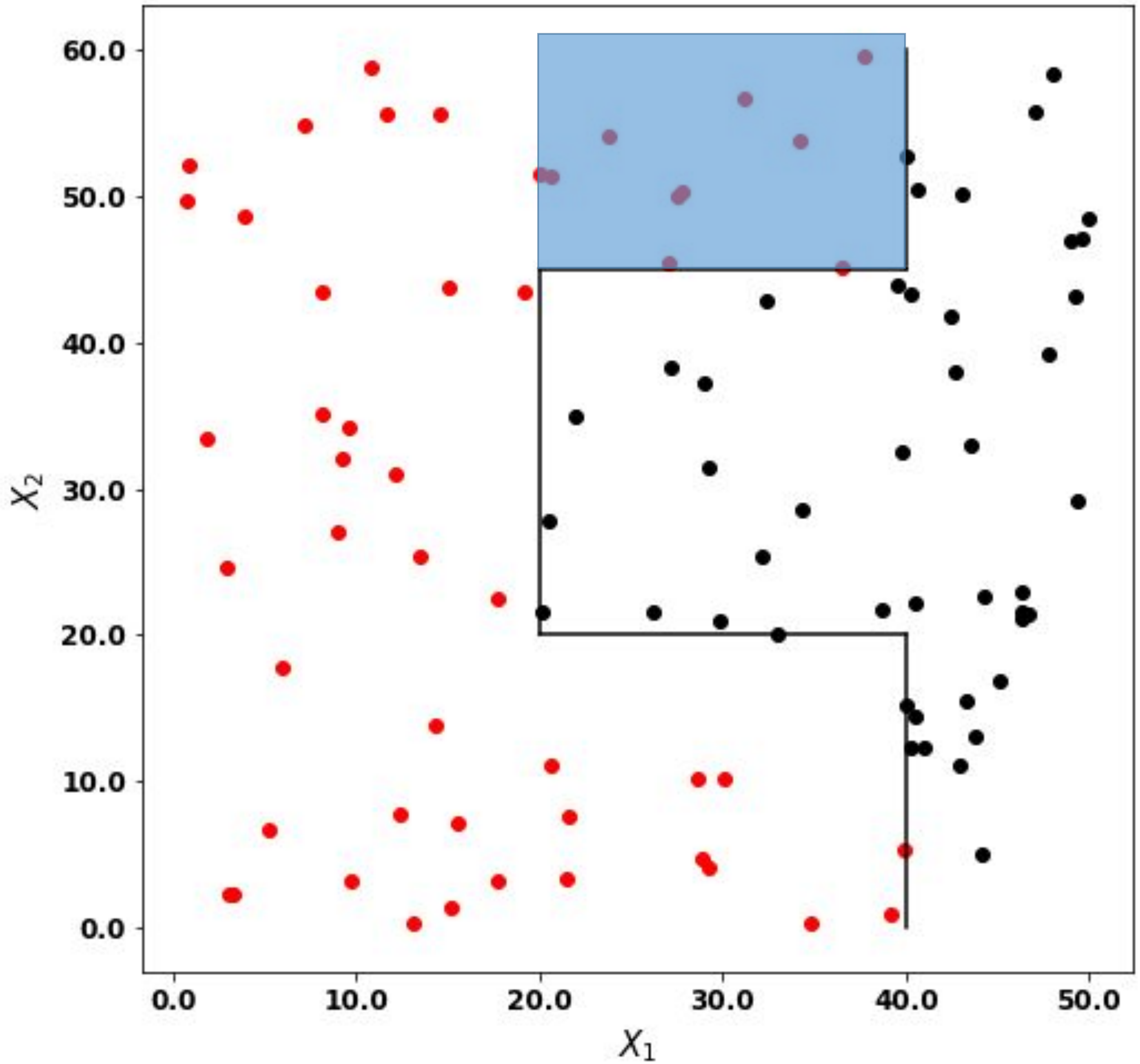
$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes
$X_2 > 45$
No
Yes
$X_2 > 20$
No
Yes

Red or black?

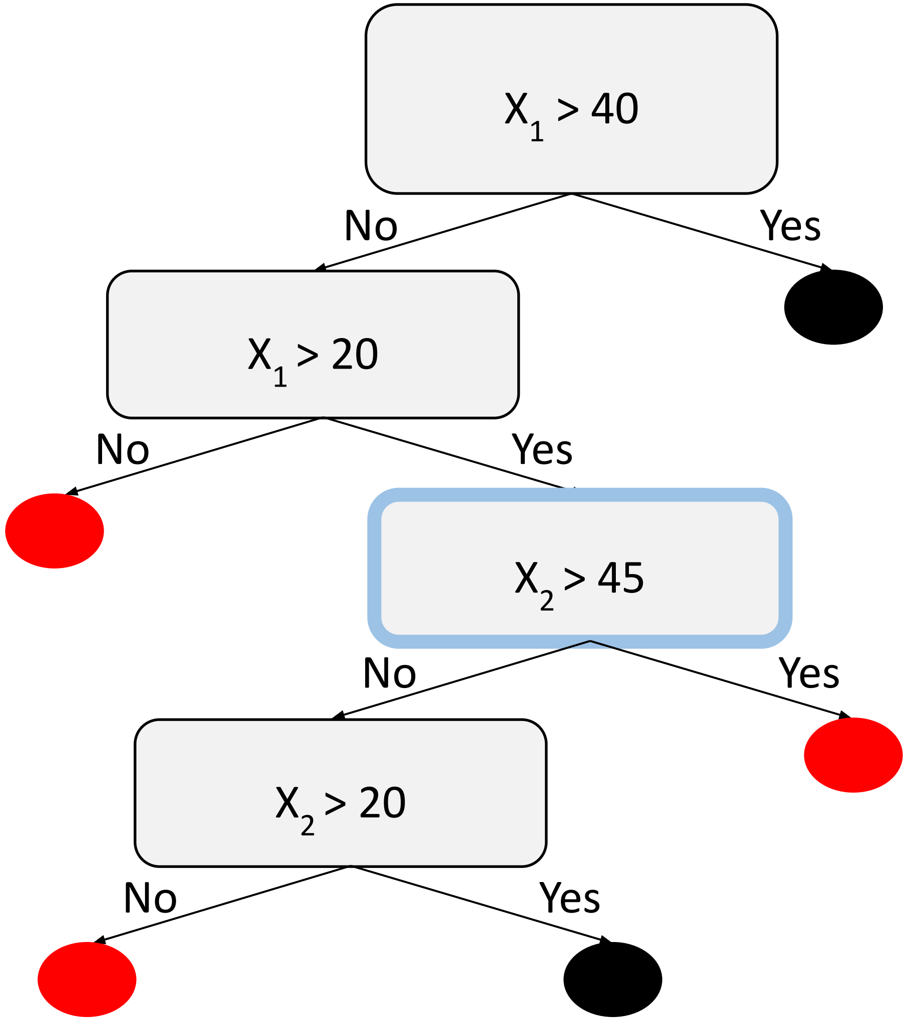

60.0
50.0
40.0
30.0
20.0
10.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
X_1
X_2


$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes
$X_2 > 45$
No
Yes
$X_2 > 20$
No
Yes

Red or black?

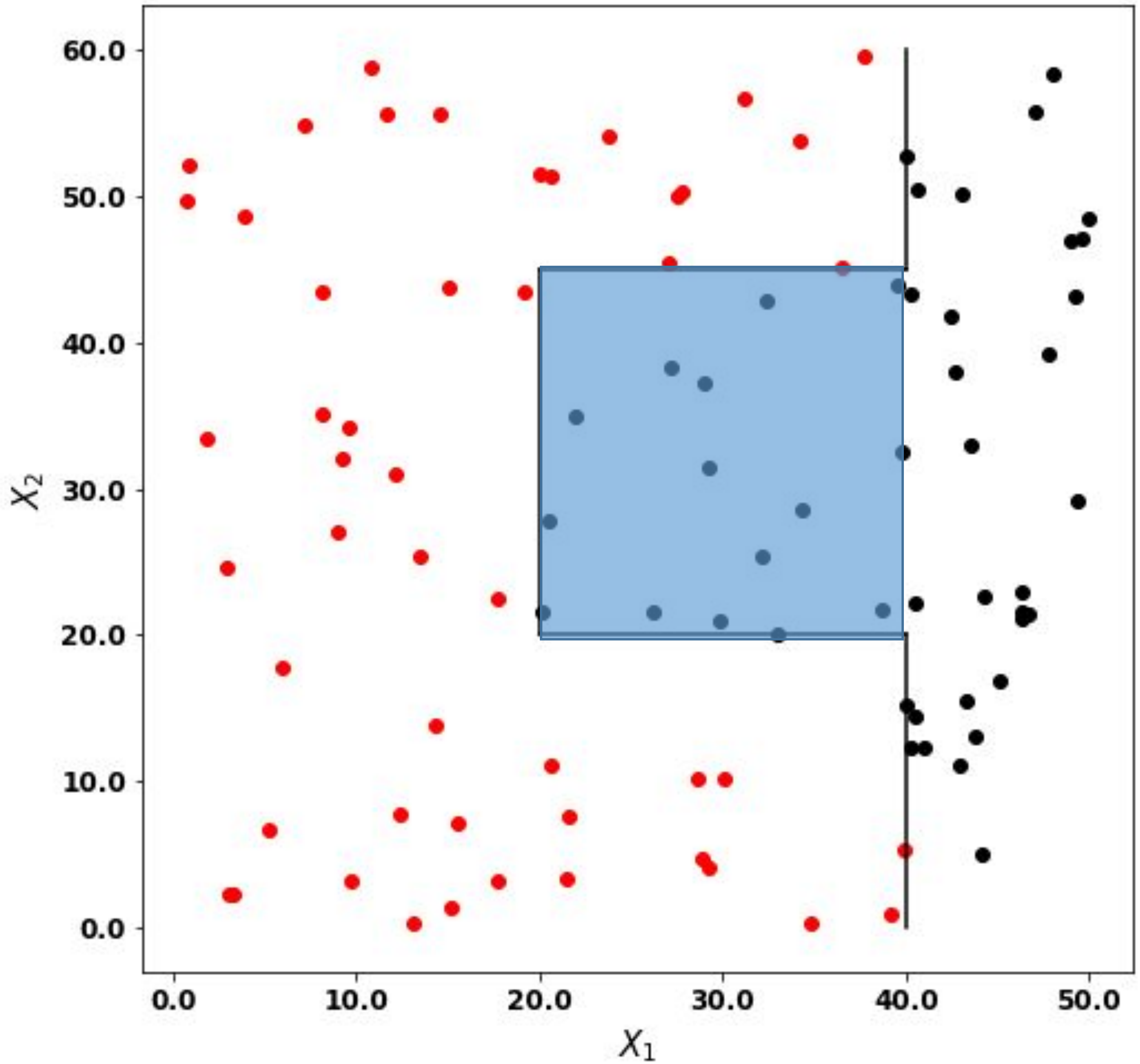


60.0
50.0
40.0
30.0
20.0
10.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
X_2
X_1

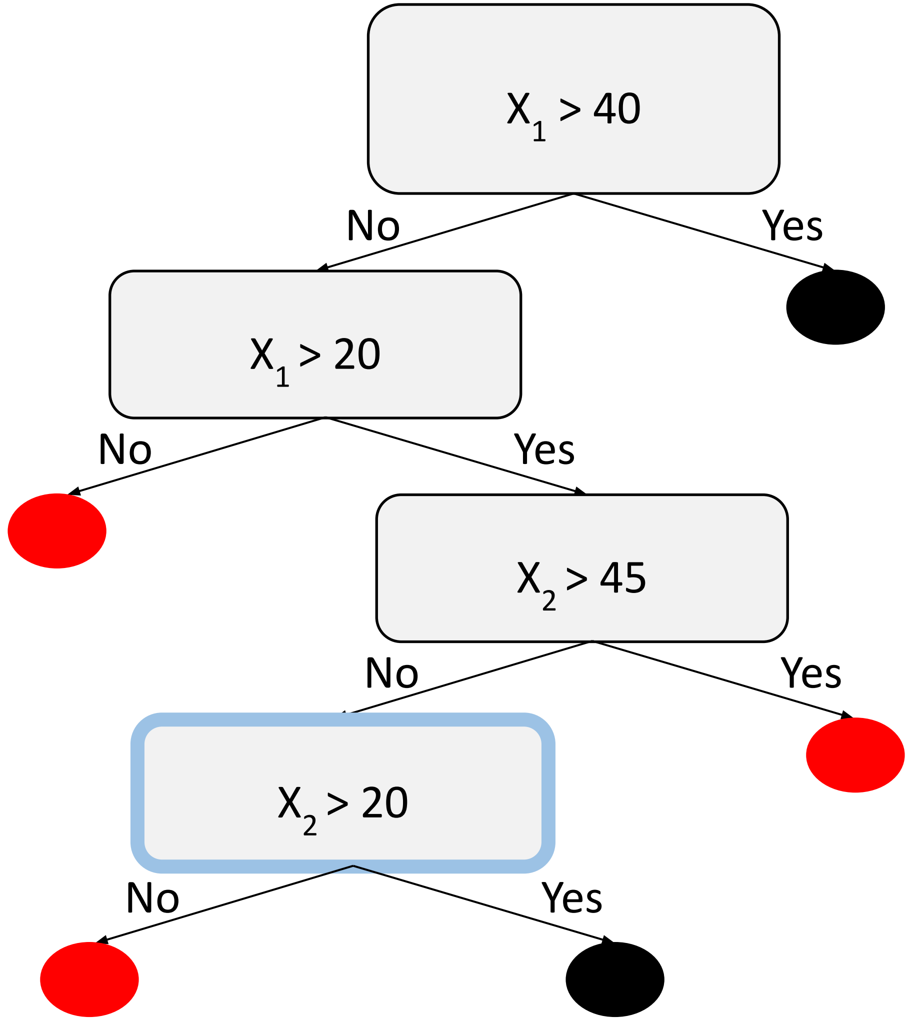


$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes
$X_2 > 45$
No
Yes
$X_2 > 20$
No
Yes

Red or black?



60.0
50.0
40.0
30.0
20.0
10.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
X_1
X_2



$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes
$X_2 > 45$
No
Yes
$X_2 > 20$
No
Yes

Decision trees for classification task

- Splits the entire feature space into small regions.
- Small regions (or leaf nodes) are tagged with the class that has most abundant training observations.

Decision tree algorithm:

- Begin with an complete feature space (not split).
- Choose an optimal* observation from an optimal feature.
- With optimal observation as a threshold make a split.
- Continue splitting the newly obtained regions until we reach pure leaf nodes or some stopping criterion** is met.

Advantages:

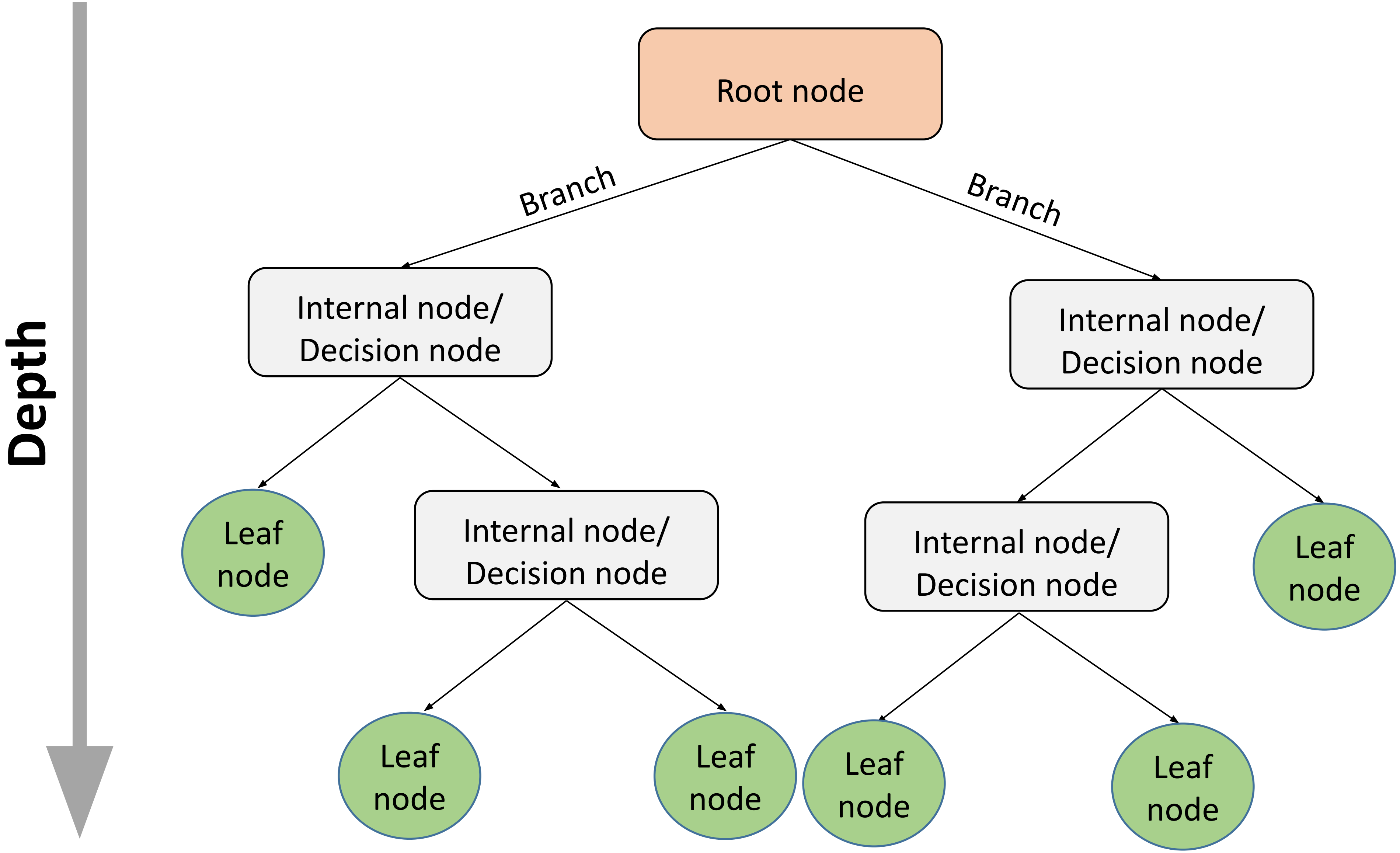
- Interpretability.
- Can handle qualitative features without pre-processing (like one-hot encoding or dummy variables).

Disadvantages:

- Not robust to change in dataset.

* Slide #9 for choice of optimal ** Slide #10 for stopping criterions

Decision trees: architecture



How to choose optimal split?

The data point of a feature that minimizes the cost function is chosen as threshold for the split.

Choice of cost function

Classification error rate	$1 - \max_k(\hat{p}_{mk})$	➢ $\hat{p}_{mk}$ ➔ Proportion of k^{th} class in training data in m^{th} region
Gini index	$\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$	➢ $\hat{p}_{mk}$ ➔ Proportion of k^{th} class in training data in m^{th} region ➢ K ➔ Total types of classes
Entropy	$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$	➢ $\hat{p}_{mk}$ ➔ Proportion of k^{th} class in training data in m^{th} region ➢ K ➔ Total types of classes

What are stopping criterion?

Maximum depth of the tree	Maximum depth till which the tree can grow
Minimum samples split	Minimum number of samples to be there in an node for a split
Minimum samples leaf	Minimum number of samples that has to be there in leaf node
Maximum leaf nodes	Maximum number of leaf nodes
Minimum impurity decrease	A node will be split only if it reduces the impurity by the given fractional value

Handling Missing Values: Imputation

- Filling in missing values *apriori*
  - Statistics literature
- Simple imputation
  - Mean, class conditioned mean etc.
- Full information imputation
- Multiple imputation
  - Monte-Carlo method
  - Several samples
  - Combine output of classifiers

Handling Missing Values: Imputation

- Ignore it
  - Not a good option, especially if there is a paucity of data
- Manually fill it
  - Tedious; time consuming; expensive
  - Sometimes essential – medical data
- Treat it as a special value
  - *Missing* ; *?*
- Infer it automatically

Inferring Missing Values

X	5	4	2	3	4	?	9	4	0
Y	I	I	II	II	I	II	I	I	II

- Mean 3.875
  - Truncated mean 3.667
  - Median 4
  - Mode 4
- Conditioned mean
  - On class labels – relatively cheap 1.667
  - One all known attributes – expensive; but best use of data

Inferring Missing Values

- Conditioning on all known attributes
  - Might bias data samples heavily, leading to poor performance of most machine learning algorithms
  - Regression
    - Ignore attribute if fit is "too good"
  - Expectation-Maximization
    - Iterate till a resulting model fits observed data well
    - Local optima

Handling Missing Values – Fractional instances

- Split instances with missing values into pieces
  - A piece going down a branch receives a weight proportional to the popularity of the branch
  - Weights sum to 1
- Info gain works with fractional instances
  - Use sums of weights instead of counts
- During classification, split the instance into pieces in the same way
  - Merge probability distribution using weights

Handling Missing Values – Surrogate Splits

- While learning, for every splitting attribute selected identify another attribute that has a very similar split to the selected attribute
- This is at the level of instances and not proportions
- At test time, if the selected attribute is missing, use the surrogate attribute
  - Can select multiple surrogates
- Cannot handle missing values during training, only during deployment

Pruning Trees

- Prune only if it reduces the estimated error
- Error on the training data is NOT a useful estimator *(would result in almost no pruning)*
  - *Why?*
- Use hold-out set for pruning (“reduced-error pruning”)

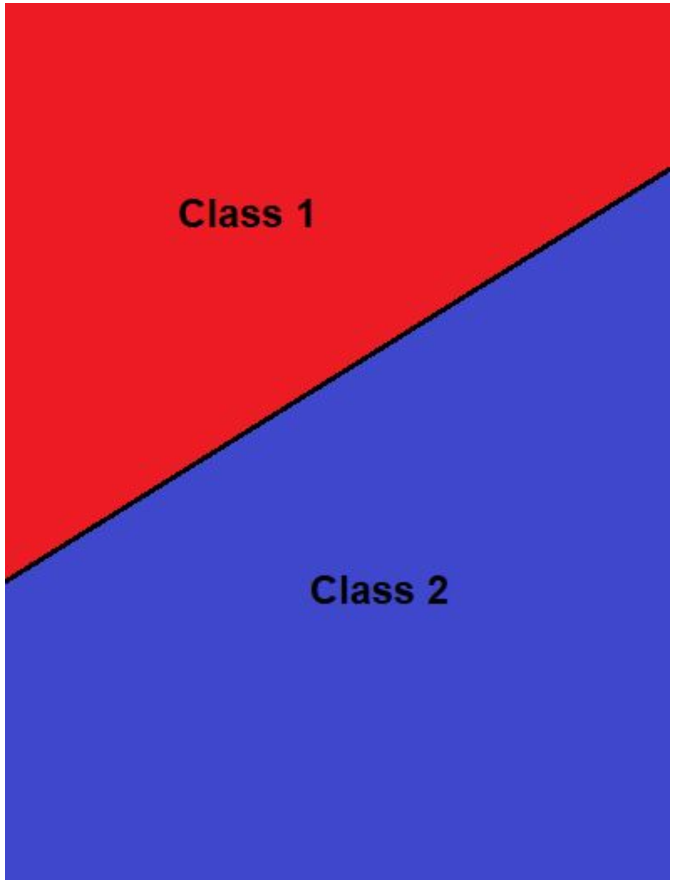
Estimating Error Rates

- CART’s method
  - Start from the lowest node in the tree
  - Compute error measure, based on error rates on **pruning** set and number of leaves
  - Compute error measure for replacing the node with a leaf having majority class label
  - If lower, then keep the replacement
- C4.5’s method
  - Derive confidence interval from **training** data
  - Use a heuristic limit, derived from this, for pruning
  - Standard Bernoulli-process-based method
  - Shaky statistical assumptions (based on training data)

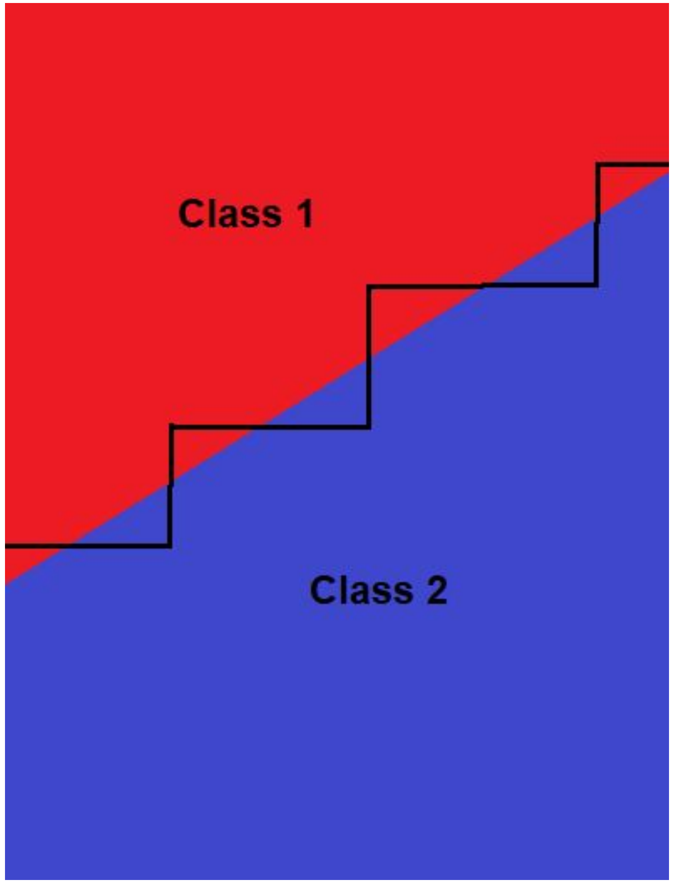
Comparing decision trees with logistic regression

Decision trees vs logistic regression: visualising the classification in dataset1

Logistic regression

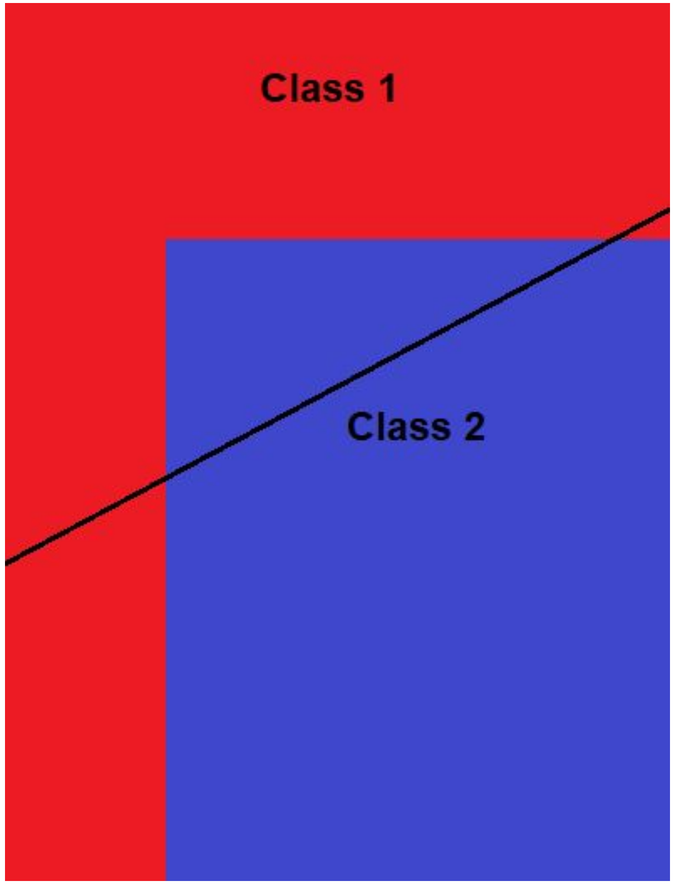


Decision trees

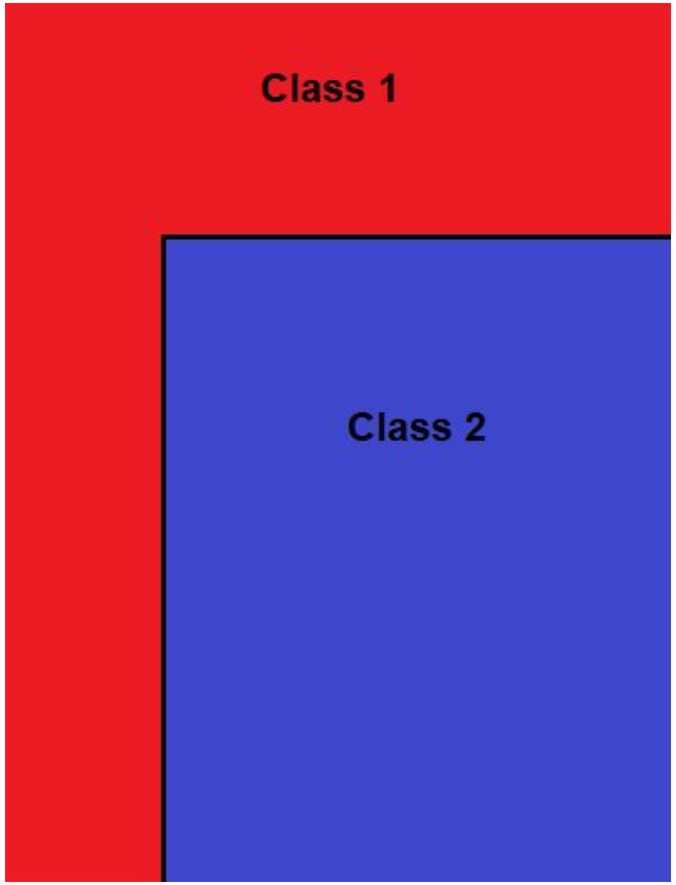


Decision trees vs logistic regression: visualising the classification in dataset2

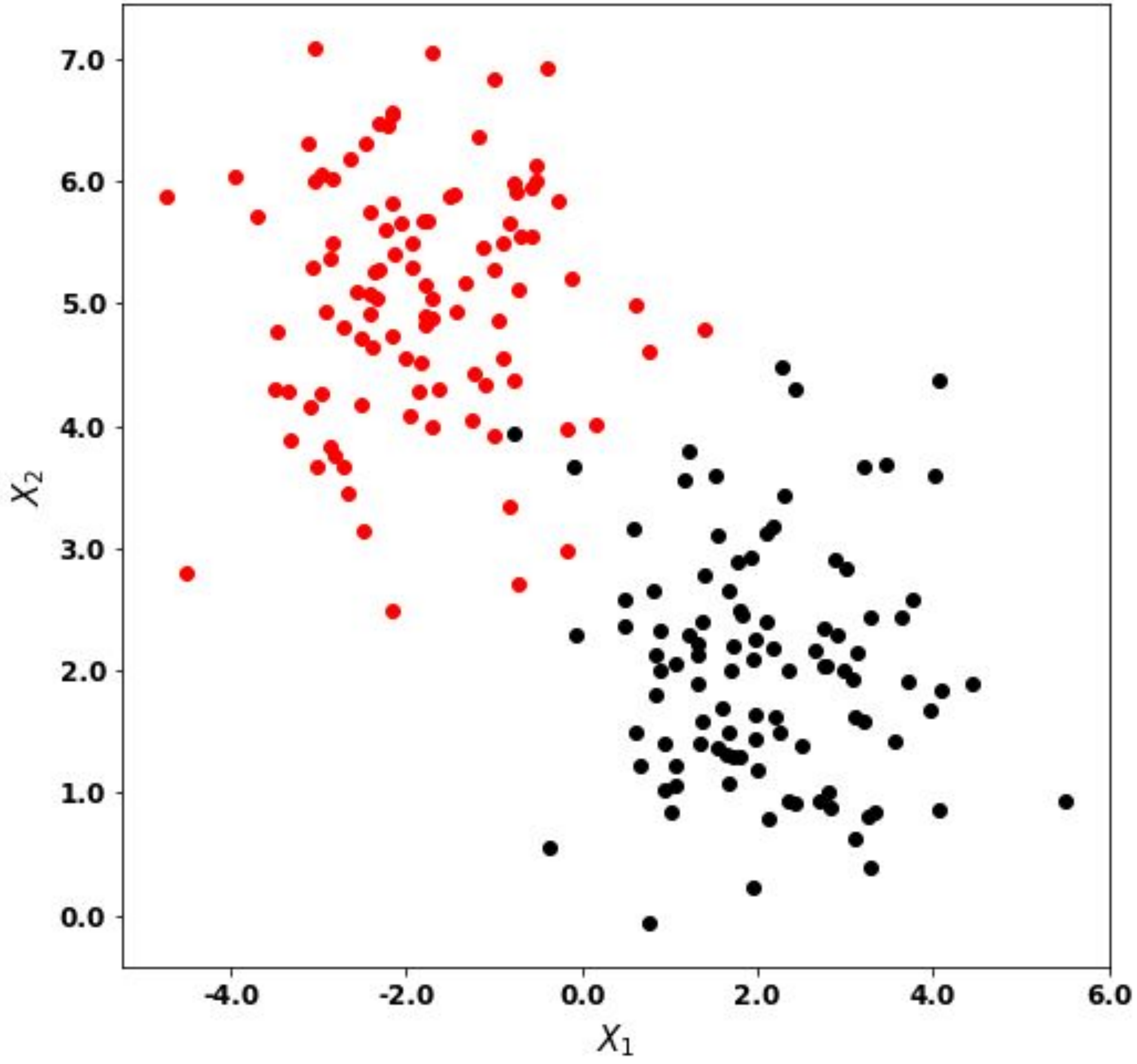
Logistic regression



Decision trees



Example dataset1



Example dataset1

Logistic regression

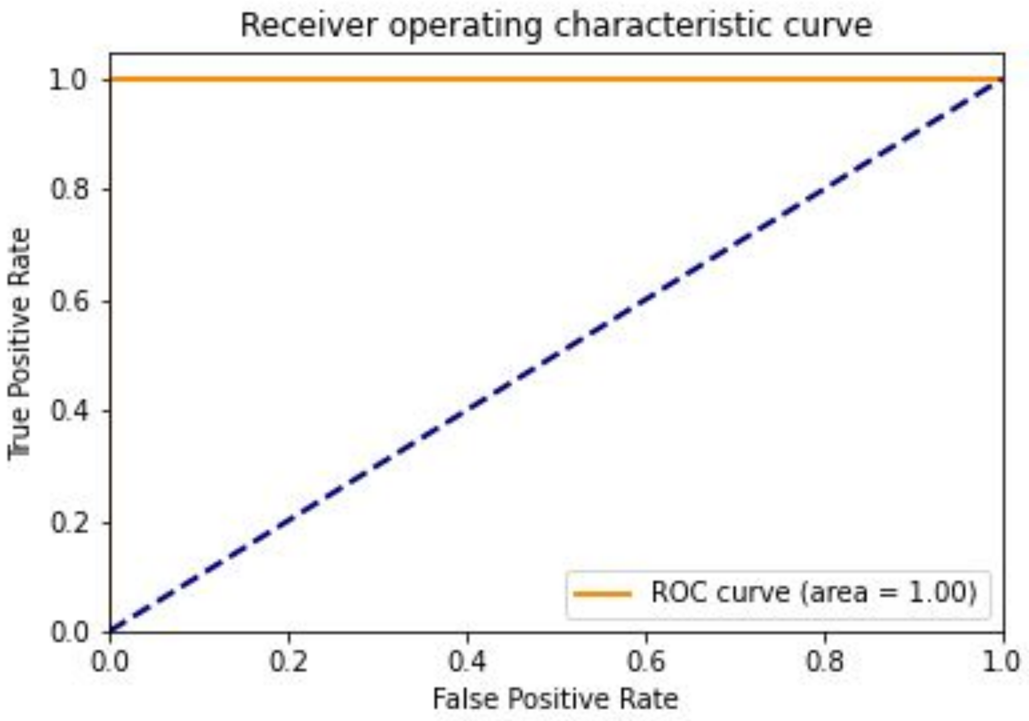
Accuracy	1
Precision	1
Recall	1
F1 score	1

Decision tree

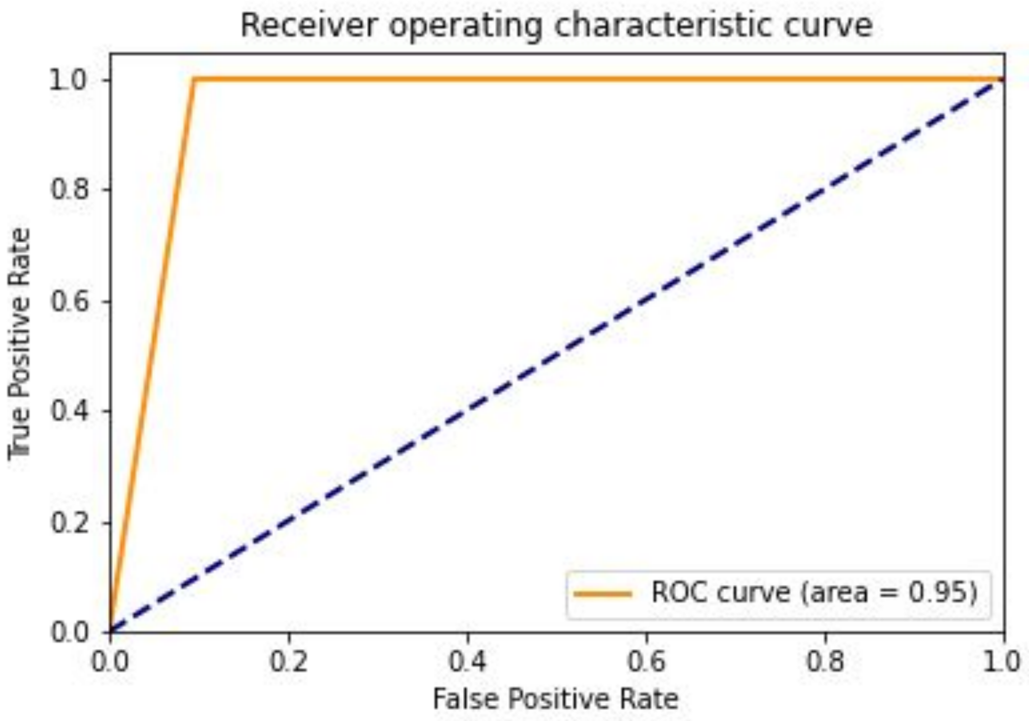
Accuracy	0.95
Precision	0.9
Recall	1
F1 score	0.95

Example dataset1

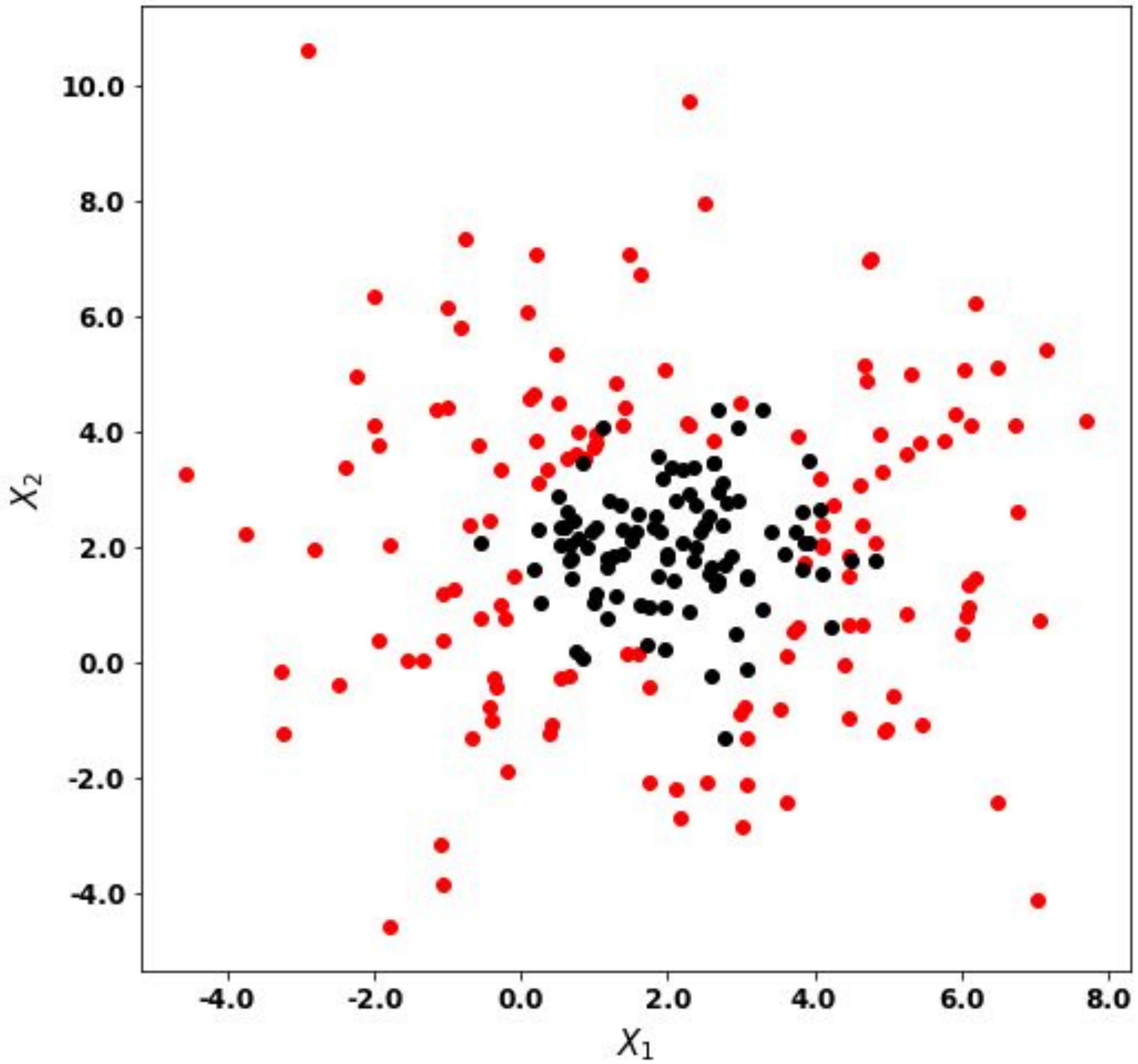
Logistic regression



Decision tree



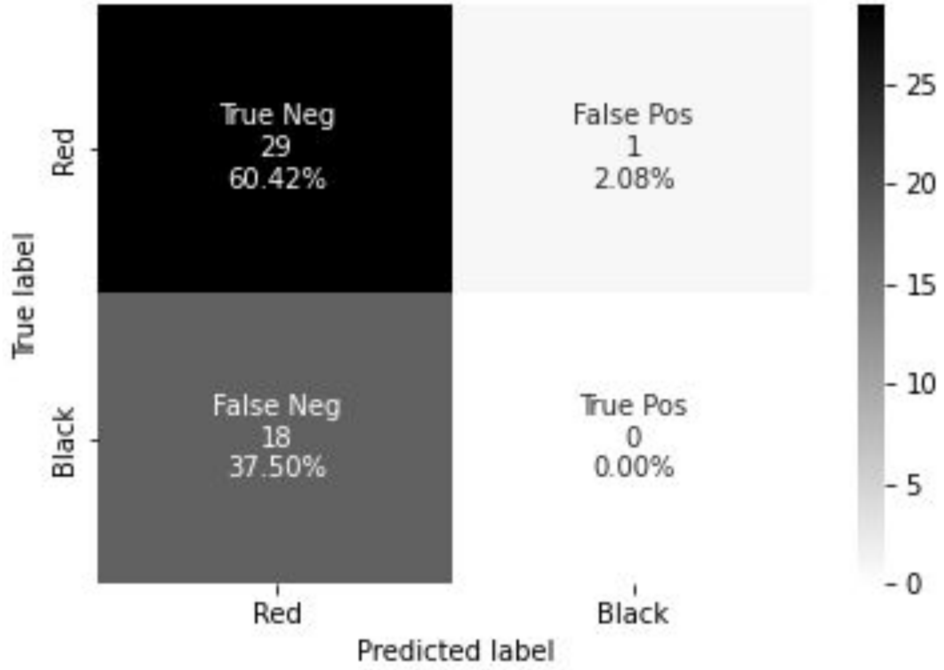
Example dataset2



Example dataset2

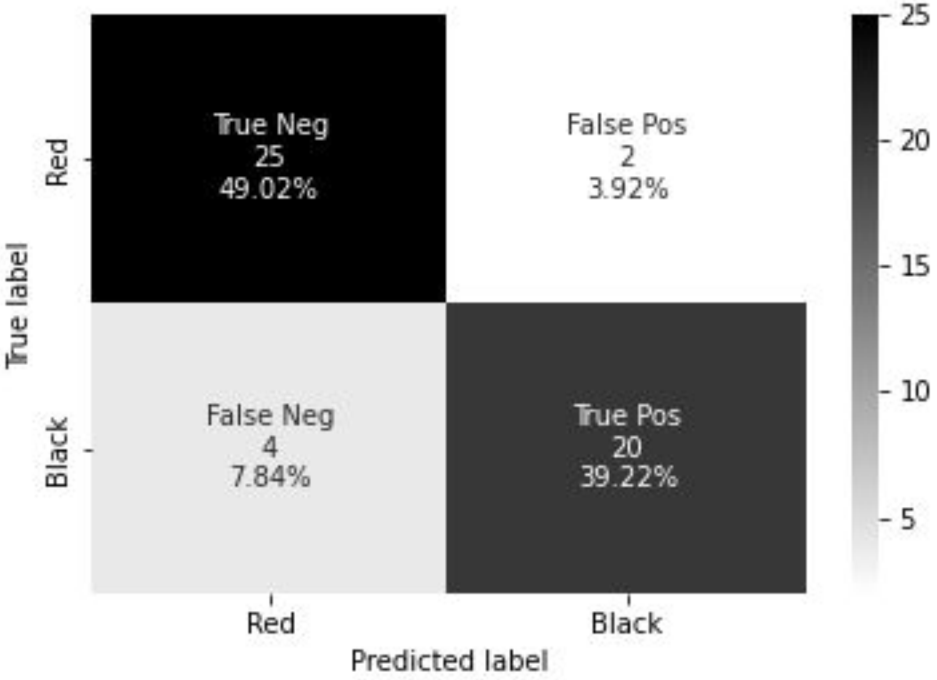
Logistic regression

Accuracy	0.6
Precision	0
Recall	0
F1 score	NaN



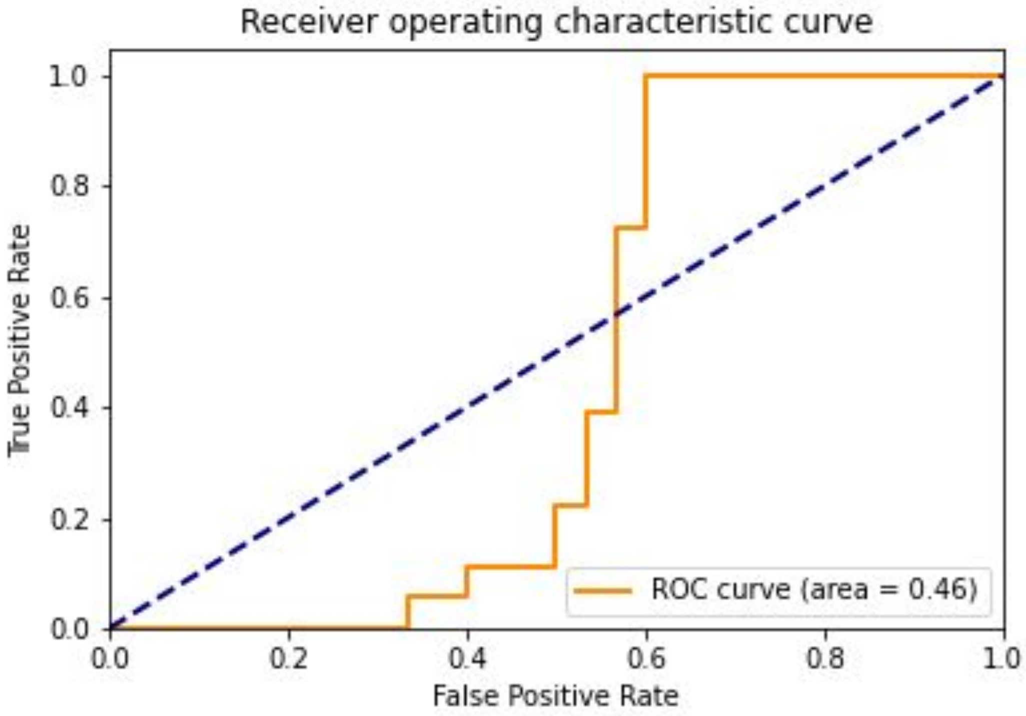
Decision tree

Accuracy	0.88
Precision	0.91
Recall	0.83
F1 score	0.87

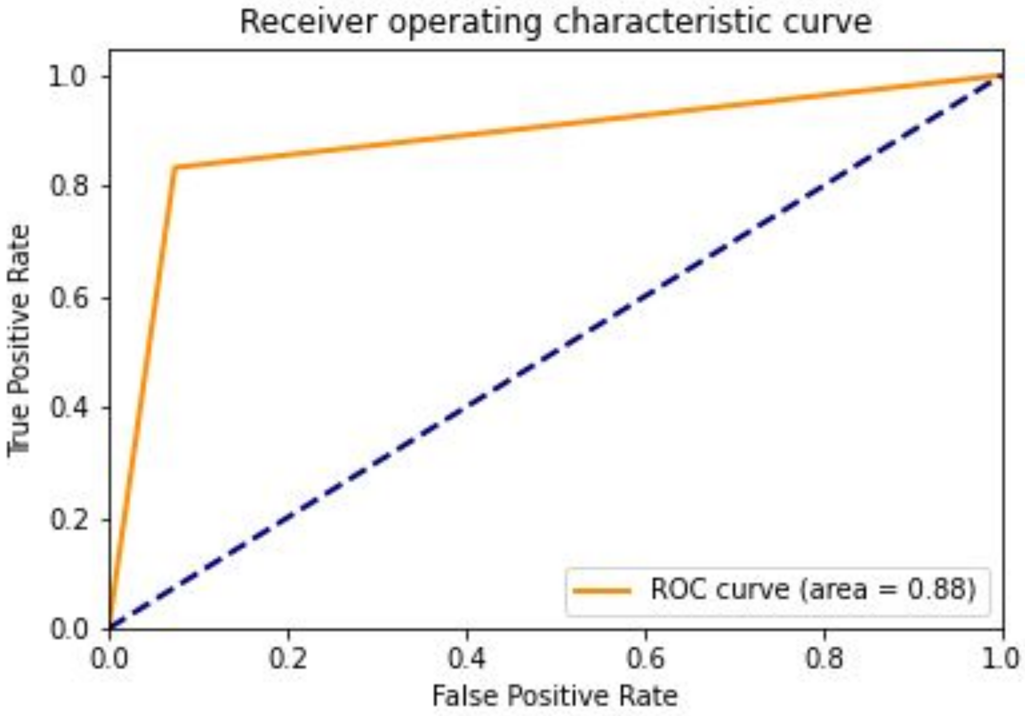


Example dataset2

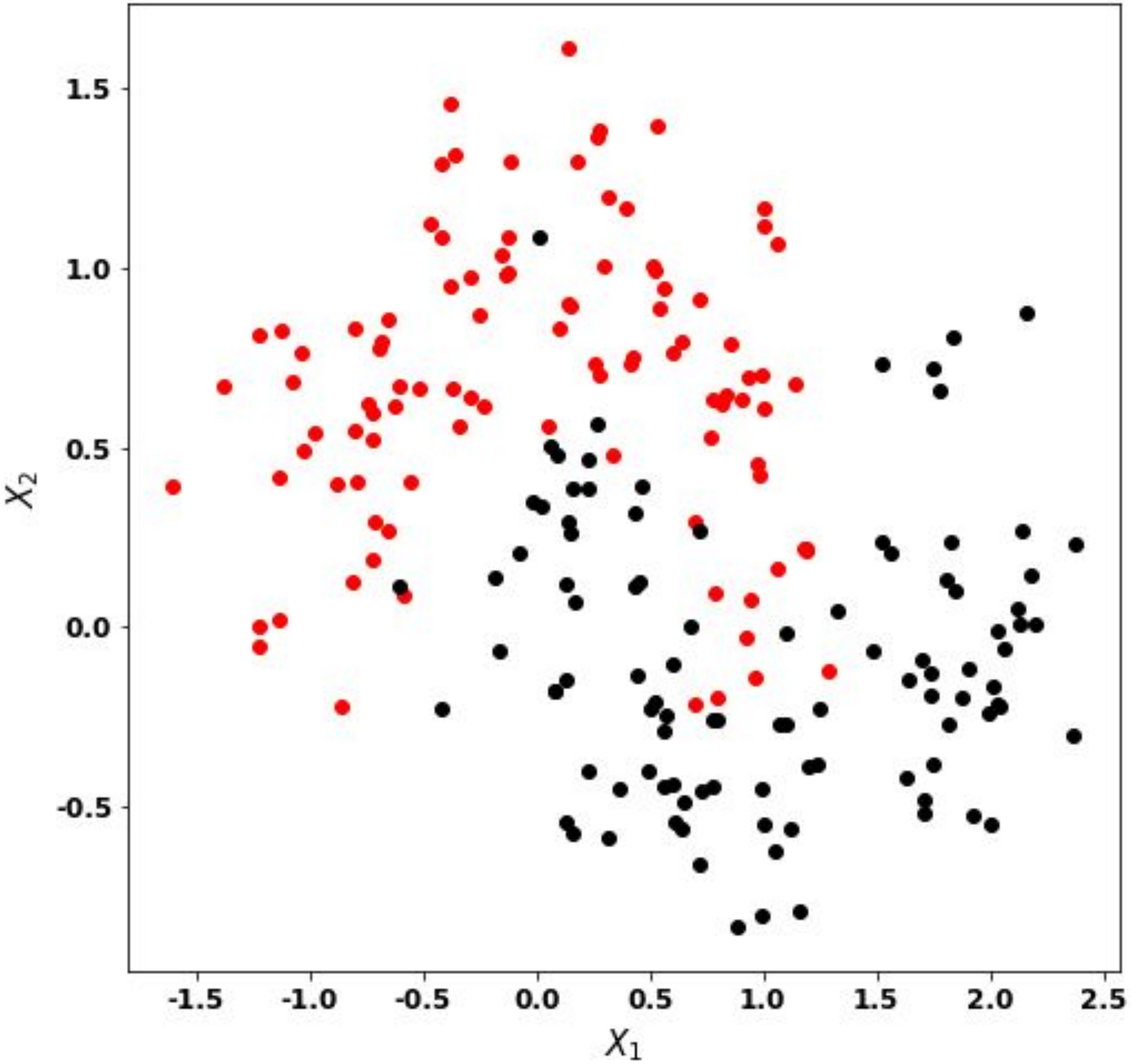
Logistic regression



Decision tree



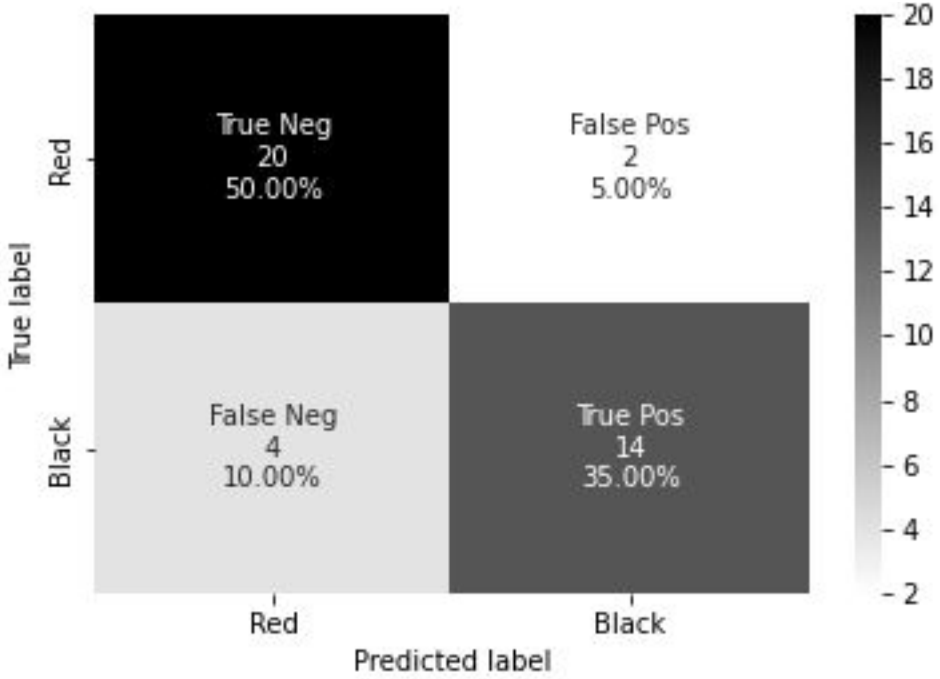
Example dataset3



Example dataset3

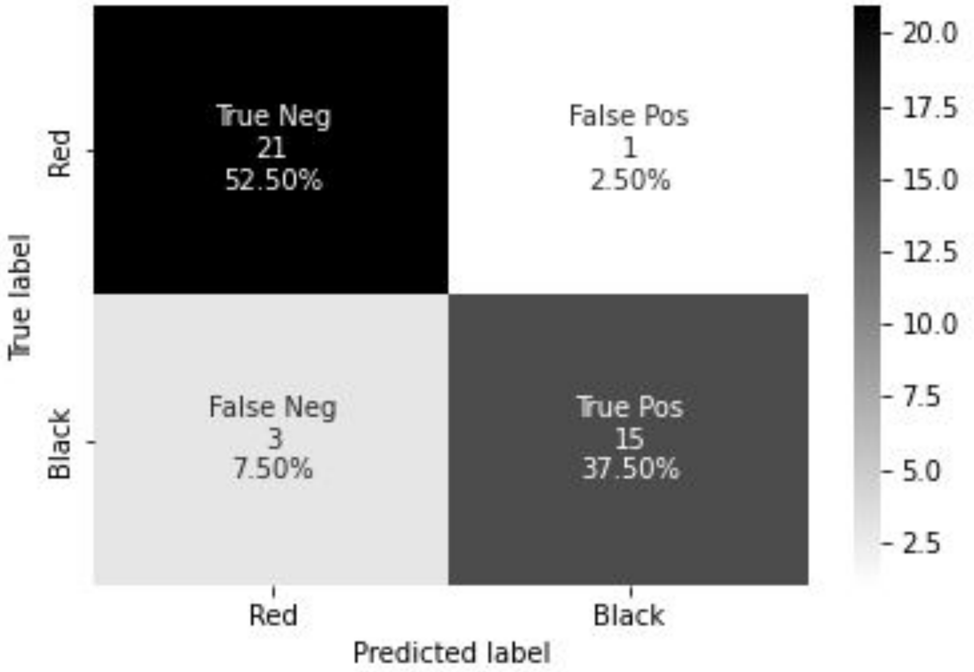
Logistic regression

Accuracy	0.85
Precision	0.88
Recall	0.78
F1 score	0.82



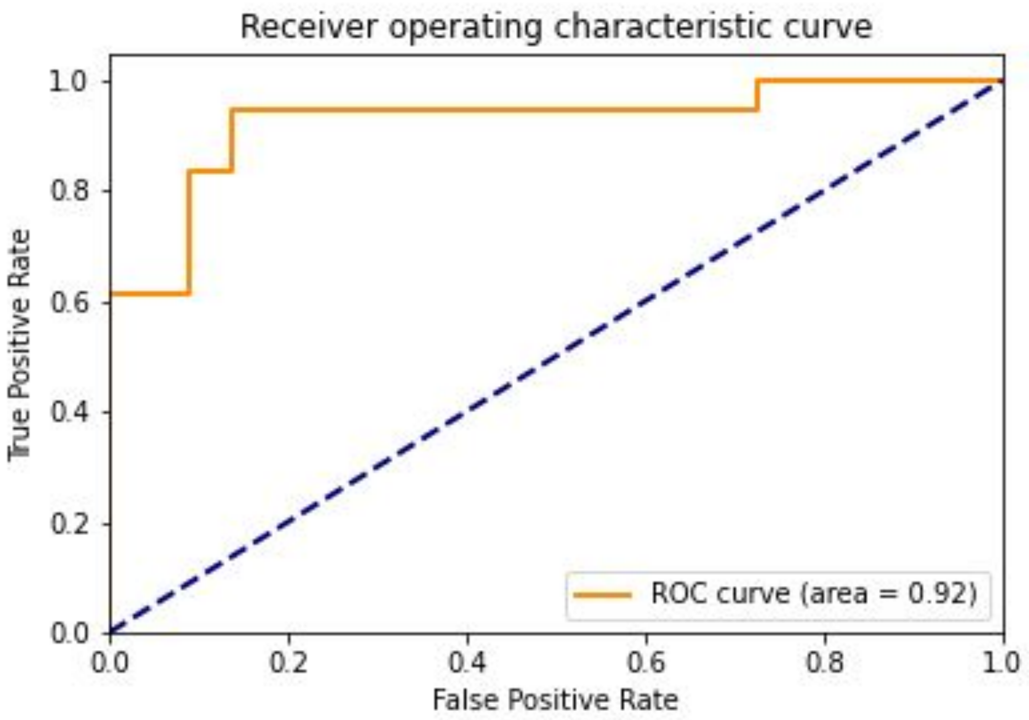
Decision tree

Accuracy	0.9
Precision	0.94
Recall	0.83
F1 score	0.88

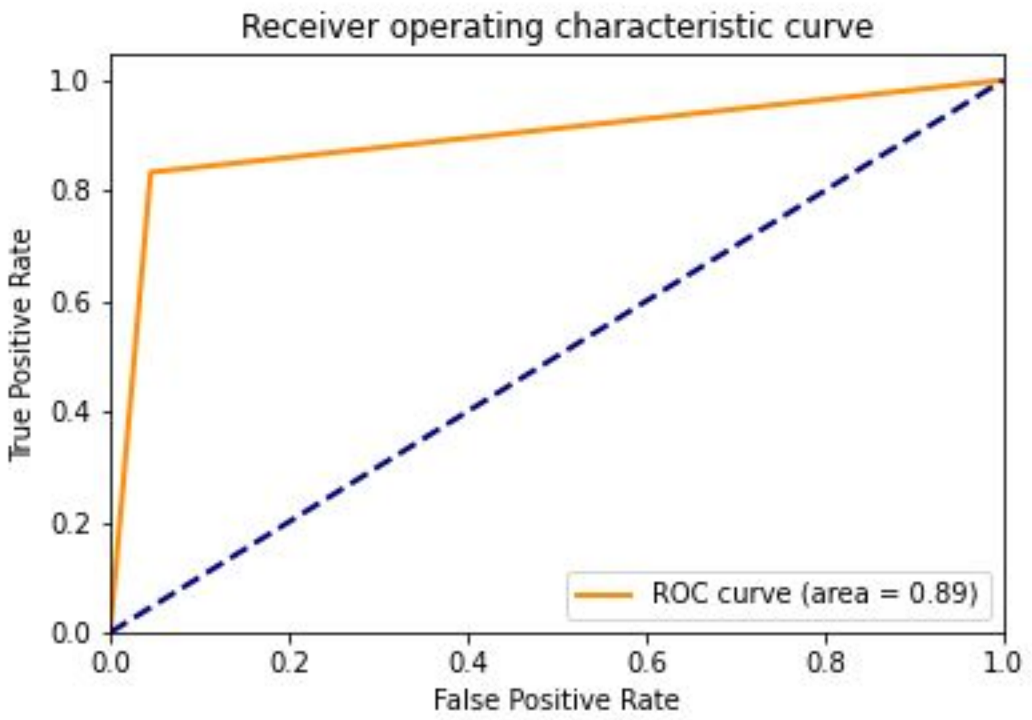


Example dataset3

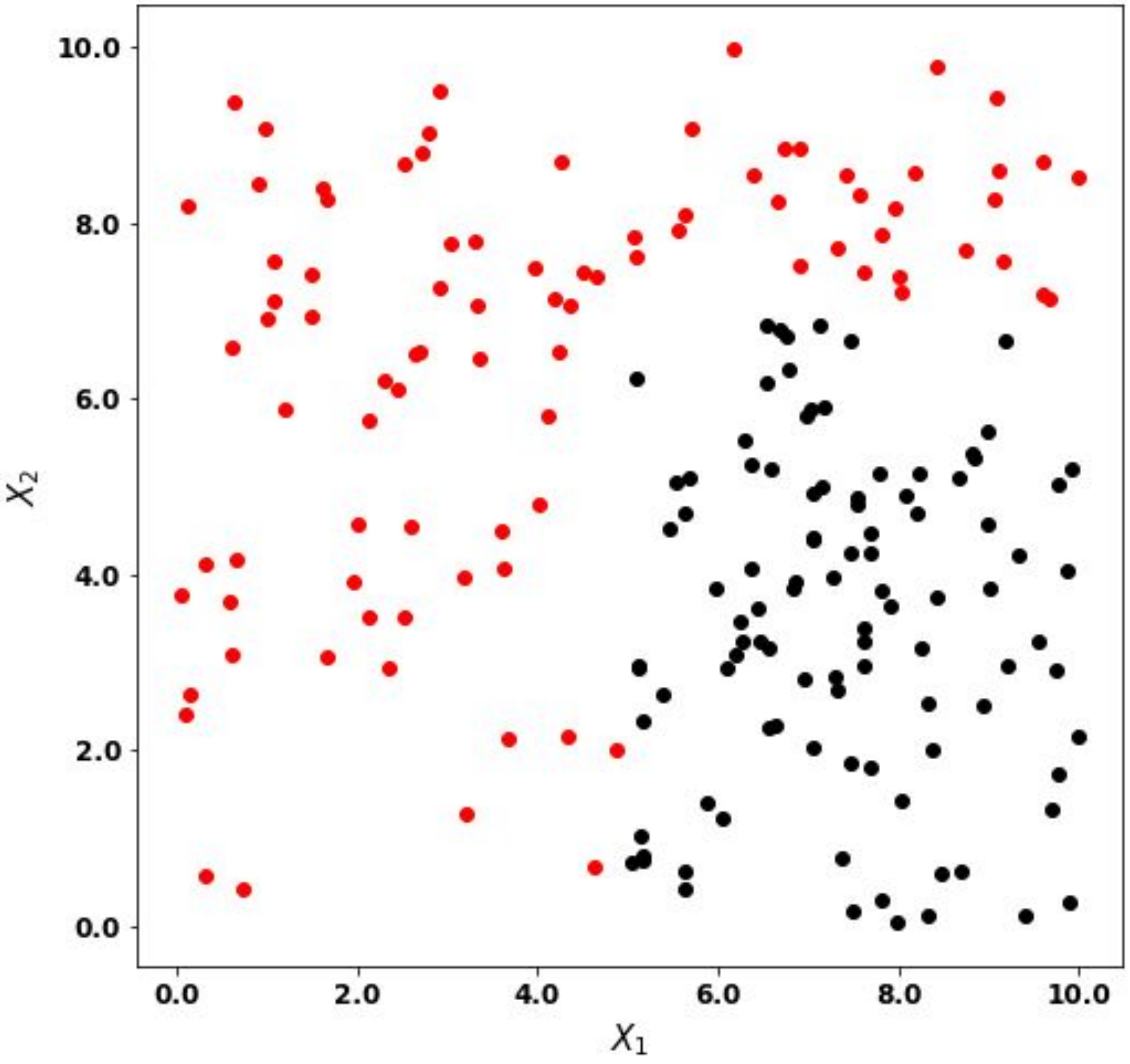
Logistic regression



Decision tree



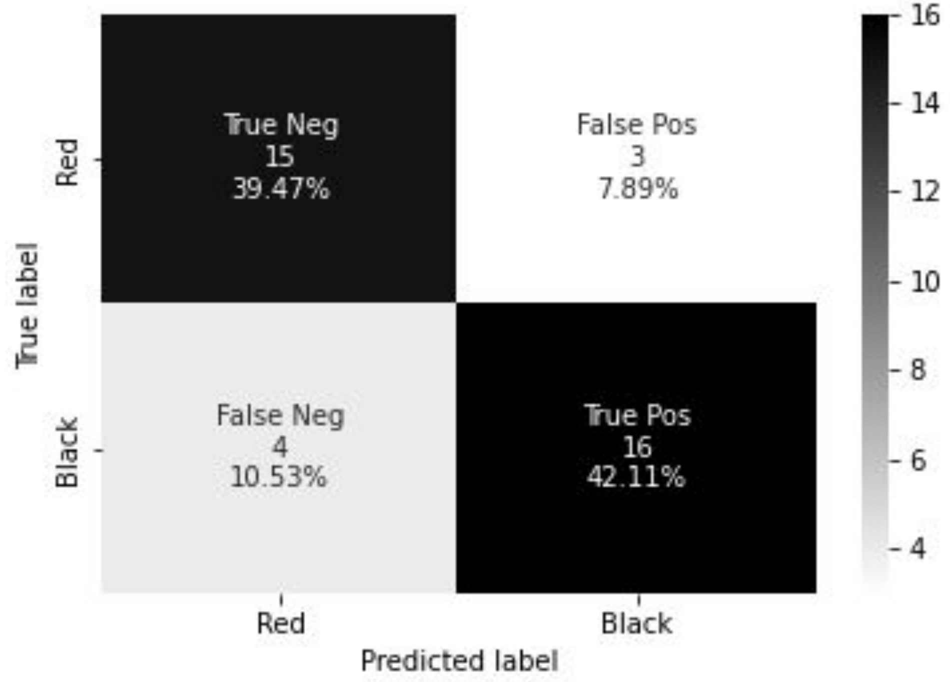
Example dataset4



Example dataset4

Logistic regression

Accuracy	0.82
Precision	0.84
Recall	0.8
F1 score	0.82

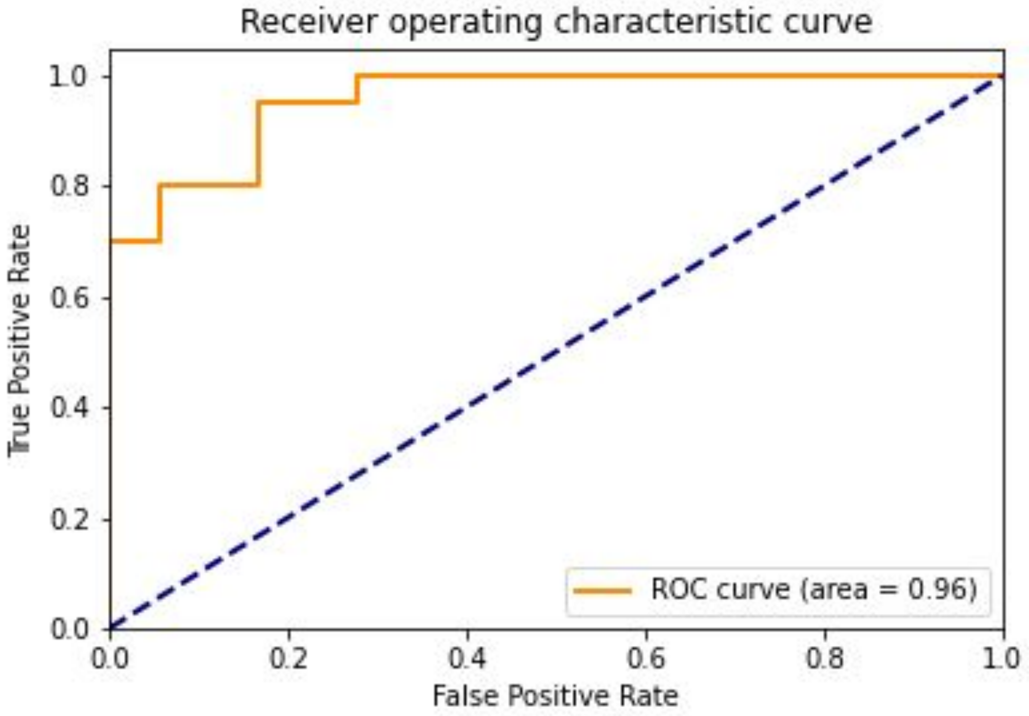


Decision tree

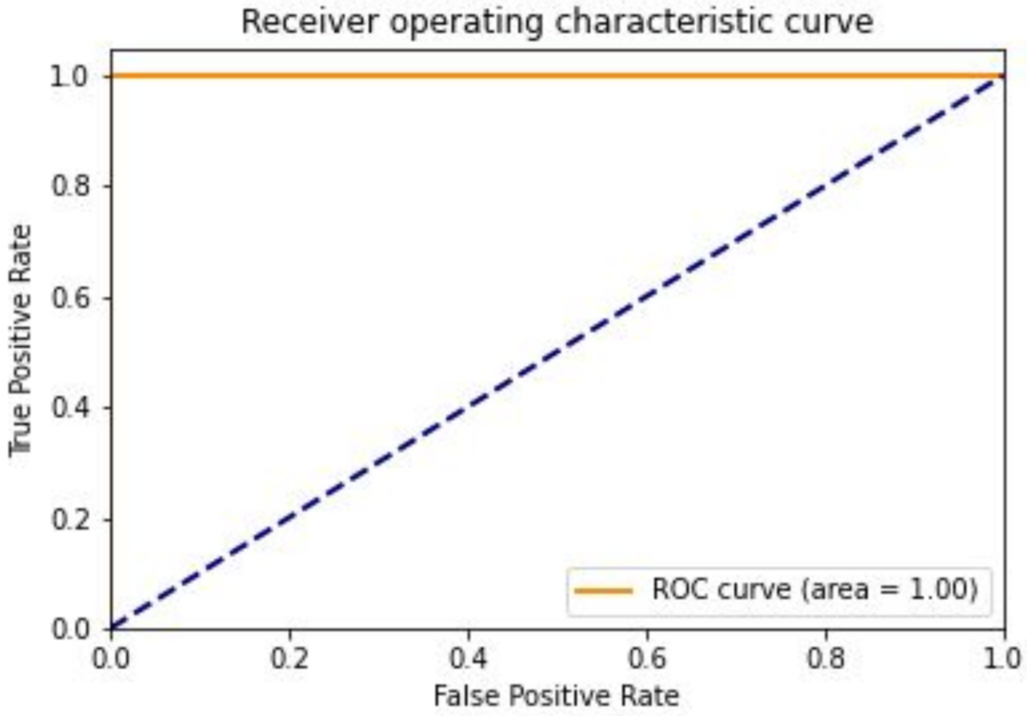
Accuracy	1
Precision	1
Recall	1
F1 score	1

Example dataset4

Logistic regression



Decision tree

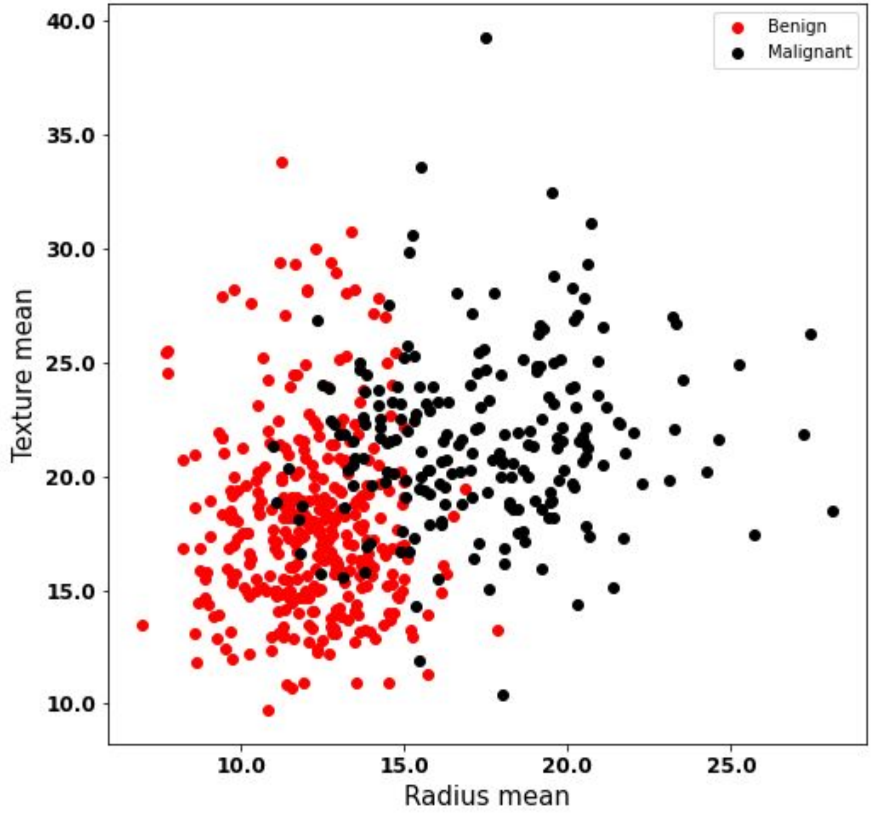


Cancer geometry data

Dataset: Geometric features of breast cancer to classify the tumour type (Benign or malignant)

Source: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

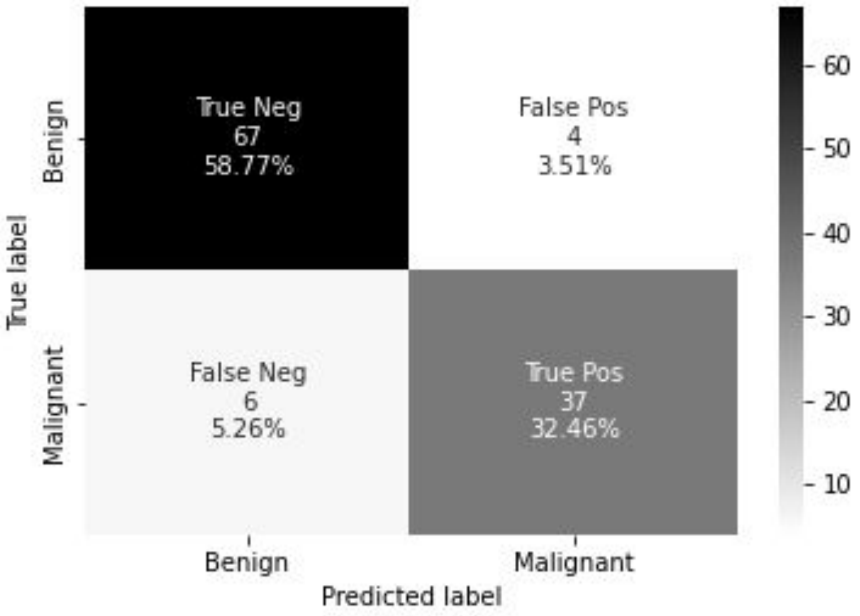
Features used for classification: radius mean and texture mean



Cancer geometry data

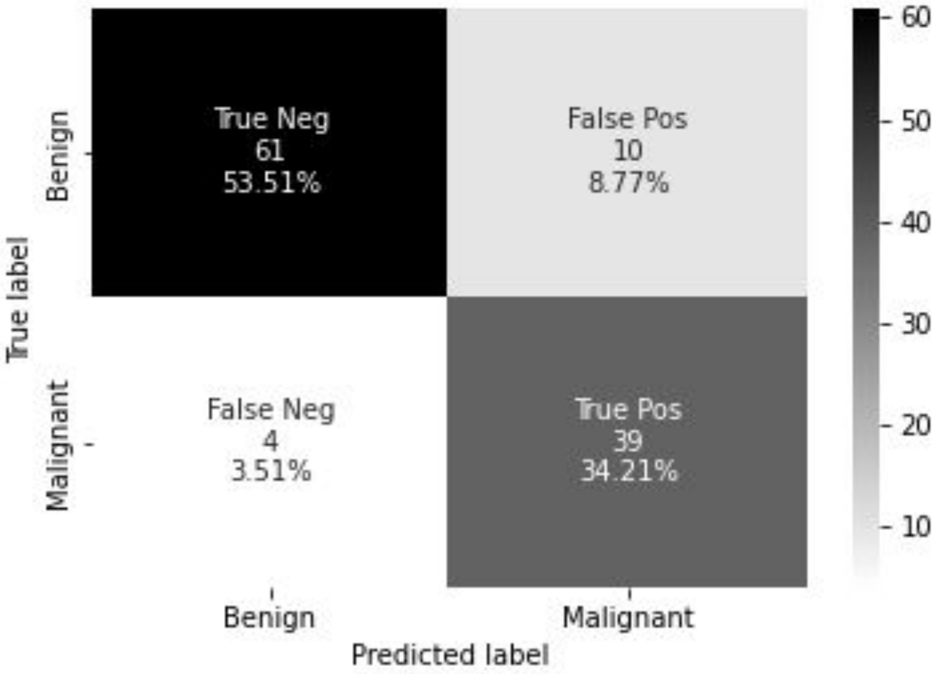
Logistic regression

Accuracy	0.91
Precision	0.90
Recall	0.86
F1 score	0.88



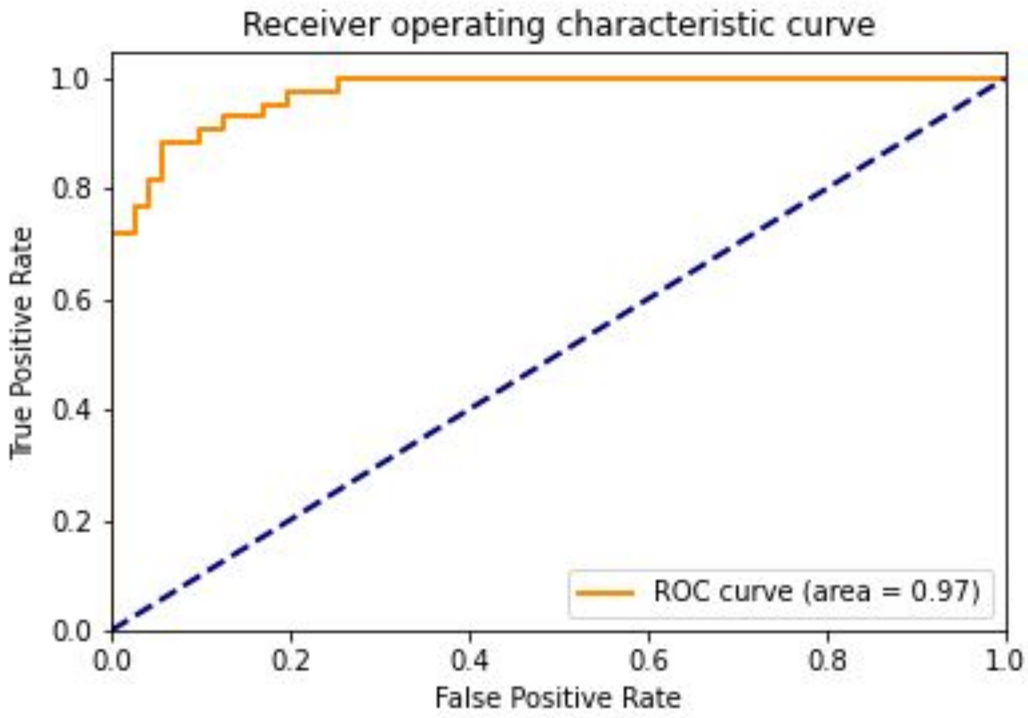
Decision tree

Accuracy	0.88
Precision	0.8
Recall	0.91
F1 score	0.85



Cancer geometry data

Logistic regression



Decision tree