Module II: Machine Learning: Foundations and Algorithms

Nirav P Bhatt
Department of Biotechnology
Robert Bosch Centre for Data Science and Artificial Intelligence
Indian Institute of Technology Madras, Chennai – 600036, India

Evaluation Metrics: Binary Classification

Confusion Matrix (Contingency table)

True \ Predicted	+	−
+	True Positive (TP)	False Negative (FN)
−	False Positive (FP)	True Negative (TN)

Accuracy = $\frac{TP+TN}{P+N}$

Precision (P) = $\frac{TP}{TP+FP}$

Recall (R)= $\frac{TP}{TP+FN}$

F1-measure = $\frac{2\ P\ R}{P+R}$

Evaluation Metrics: Binary Classification

Receiver Operating Characteristic (ROC) Curve

True Positive Rate = $\frac{TP}{TP+FN}$

False Positive Rate = $\frac{FP}{FP+TN}$

Area Under the Curve (AUC)

(ROC) Curve: A graph between FPR vs TPR

Classification Techniques

Naïve Bayes classifier

Decision Tree classifier

Probability basics: Marginal probability

Marginal probability of A (or) probability of A (or) P(A) ➔ Probability of event A happening

Consider an example case of a person jumping from height using a Parachute in India. (S)he has an equal chance of landing anywhere in India.

Event A: Landing in Tamil Nadu
Event B: Landing in Water body in Mainland India

$$P(A) = \frac{\text{Tamil Nadu}}{\text{India}}$$

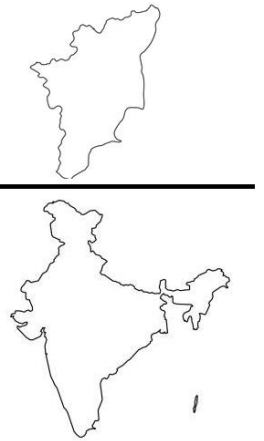


Figure 1A: P(A) is given by area of Tamil Nadu upon area of India

$$P(B) = \frac{\text{Waters in India}}{\text{India}}$$

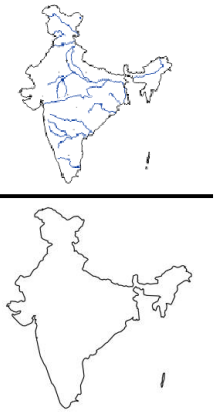


Figure 1B: P(B) is given by area of waters in India upon area of India

Probability basics: Joint probability

The joint probability of A and B (or) P(A, B) ➔ Probability of both the events A and B happening

Consider an example case of a person jumping from height using parachute in India. (S)he has equal chance of landing anywhere in India.

Event A: Landing in Tamil Nadu
Event B: Landing in Water body in Mainland India

$$P(A,B) = \frac{\text{  }}{\text{  }}$$

Figure 2: P(A,B) is given by area of waters in Tamil Nadu upon area of India

Probability basics: Conditional probability

Conditional probability of A given B (or) P(A|B) ➔ Probability of the event A happening given that event B has already happened

Consider an example case of a person jumping from height using a parachute in India. (S)he has an equal chance of landing anywhere in India.

Event A: Landing in Tamil Nadu
Event B: Landing in Water body in Mainland India

$$P(A|B) = \frac{\text{  }}{\text{  }}$$

Figure 3: P(A|B) is given by area of waters in TamilNadu upon area of waters in India

Bayes theorem

Posterior

Likelihood

Prior

$$P(A|B_1, B_2) = \frac{P(B_1,B_2|A)P(A)}{P(B_1,B_2)}$$

Marginal

Event A: Landing in Tamil Nadu

Event B: Landing in Waters

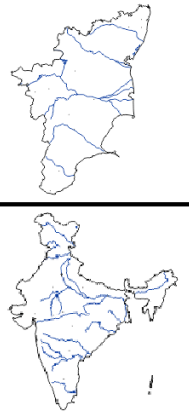
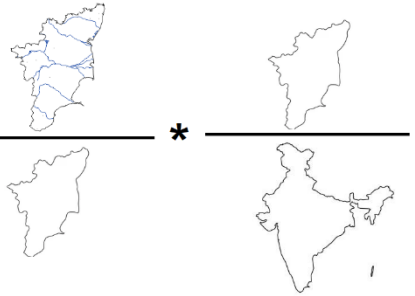
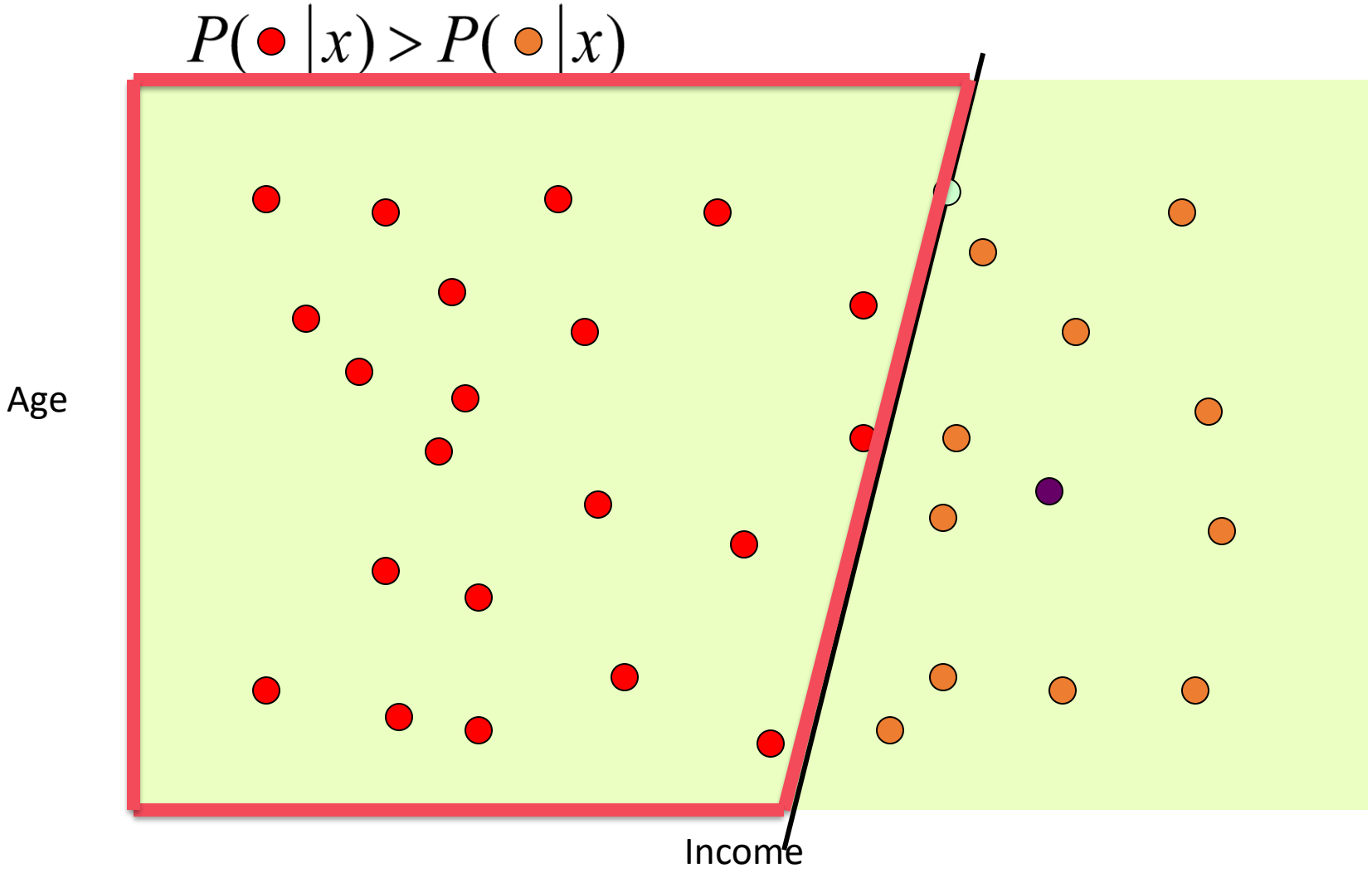
$$P(A|B) = \frac{\text{  }}{} = \frac{P(B|A)P(A)}{P(B)} = \frac{\text{  } * \text{}}{\text{  }}$$

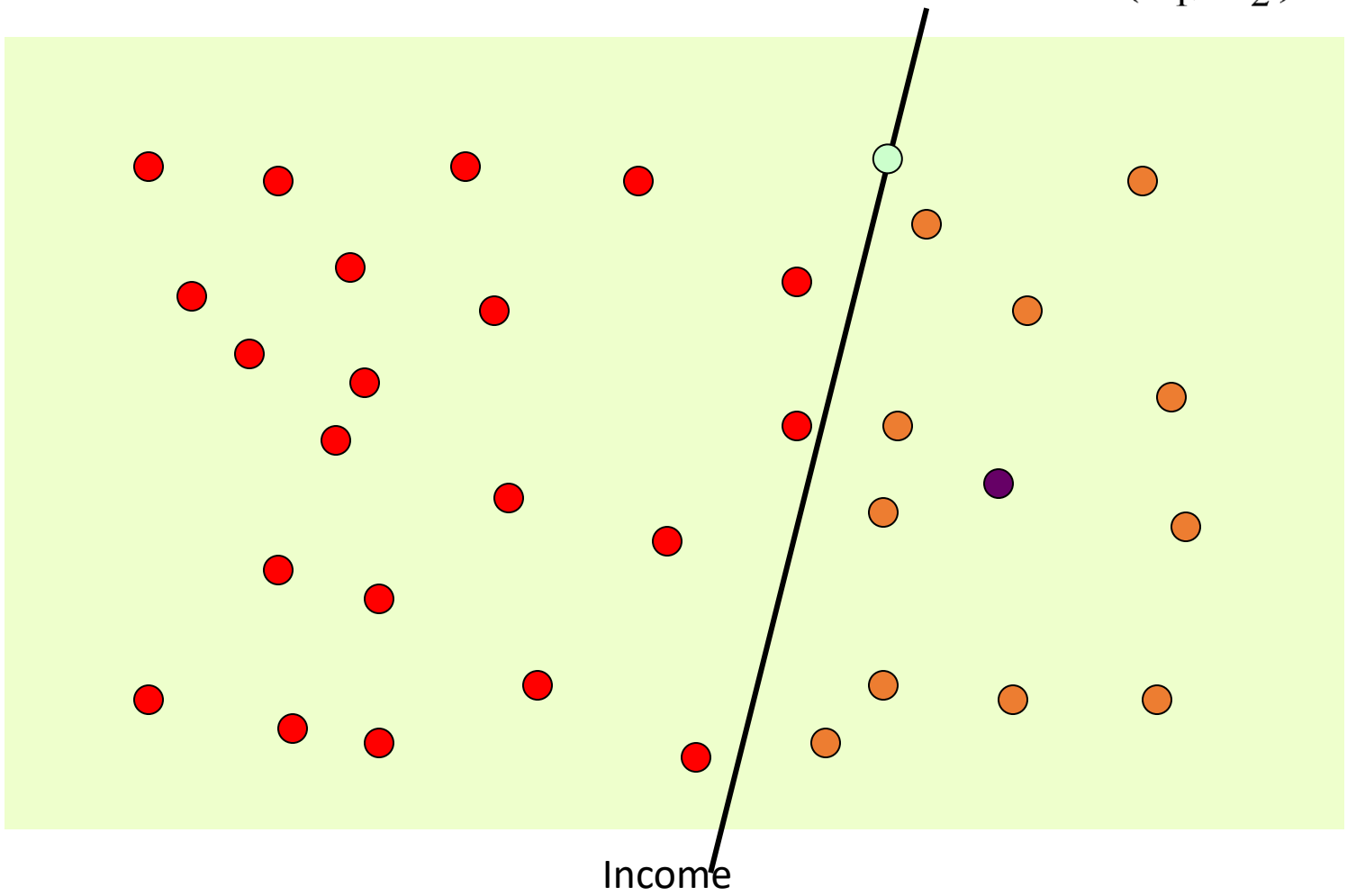
Figure 4: Implication of Bayes theorem to get probability of A given B

Separating Hyperplane



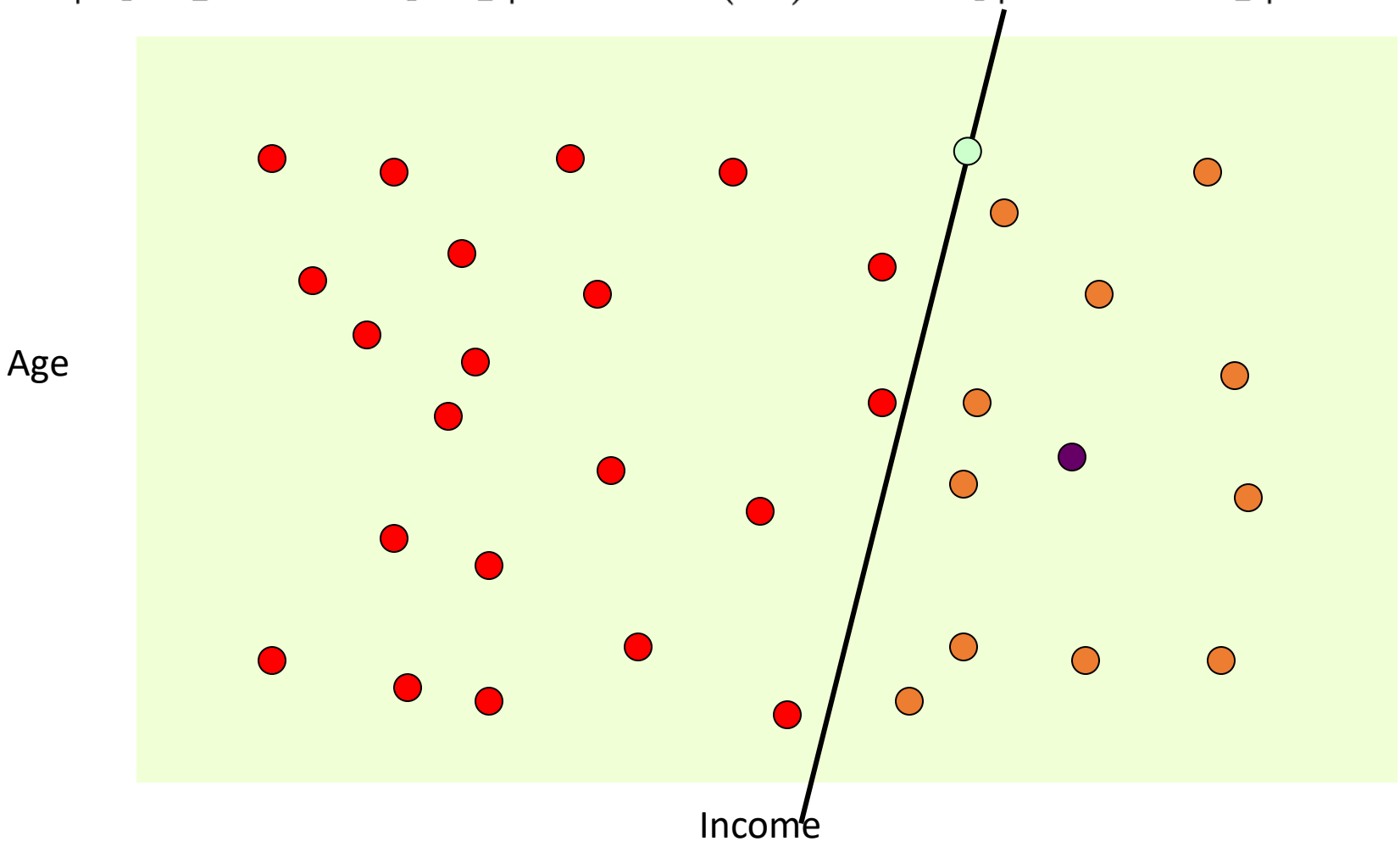
Bayes Theorem

$$P(\bullet \mid x) = P(\bullet \mid x_1, x_2) = \frac{P(x_1, x_2 \mid \bullet) \times P(\bullet)}{P(x_1, x_2)}$$



Naïve Bayes

$$P(\bullet \mid x_1, x_2) \propto P(x_1, x_2 \mid \bullet) \times P(\bullet) = P(x_1 \mid \bullet) \times P(x_2 \mid \bullet) \times P(\bullet)$$



Naïve Bayes classifier

- Assumption: The features are independent given the class labels
- Simple form for the probability distribution
- Not necessarily linear hyperplane 😊.
- Typically estimate by counting co-occurrences of feature value with class label
  - Maximum likelihood estimate
- Surprisingly powerful, especially in data with many features
  - High dimensional spaces

Naïve Bayes classifier

Bike Name	Weight	Engine
Y	100	300
Y	110	250
Y	92	250
Y	80	200
S	90	250
S	65	200
S	80	150
S	70	175

Predict the bike that was purchased from a given set of features,

Weight = 85 and engine = 250, Bike = ??

Continuous and Independent variable

$P(Y) = P(S) = 0.5$

Naïve Bayes classifier

Bike Name	Weight	Engine
Y	100	300
Y	110	250
Y	92	250
Y	80	200
S	90	250
S	65	200
S	80	150
S	70	175

objective

$P(? \mid W=85, E=250)$

$W, E \sim$ Gaussian distribution

$W \sim N(\mu_W, \sigma_W^2)$

$E \sim N(\mu_E, \sigma_E^2)$

Bike	W	E
Y	(95.5, 161)	(250, 1666.67
S	(76.25, 122.9)	(193.7, 1822.9)

Naïve Bayes classifier

$$P(Y \mid W=85, E=250) = \frac{P(W=85, E=250 \mid Y)\,P(Y)}{P(W=85, E=250)}$$

$$= \alpha\, P(W=85 \mid Y) \cdot P(E=250 \mid Y)$$

W & E : Gaussian.

$$P(x = x_1 \mid c) = \frac{1}{\sqrt{2\pi\sigma_x^2}}\, e^{-\frac{(x_1 - \mu_x)^2}{2\sigma_x^2}}$$

Naïve Bayes classifier

$P(W=85|Y) = 0.022$ | $P(W=85|S) = 0.026$

$P(E=250|Y) = 0.0098$ | $P(E=250|S) = 0.0039$

$P(Y|W=85, E=250) > P(S|W=85, E=250)$

Y is the class.

Naïve Bayes classifier

- In a given dataset, let X_1, X_2, X_3,…X_n be the features and $Y \in [0, 1]$ be the class to be predicted. Applying Bayes theorem,

$$P(Y|X_1, X_2,\ldots, X_n) = \frac{P(X_1,X_2,\ldots,X_n|Y)P(Y)}{P(X_1,X_2,\ldots,X_n)}$$

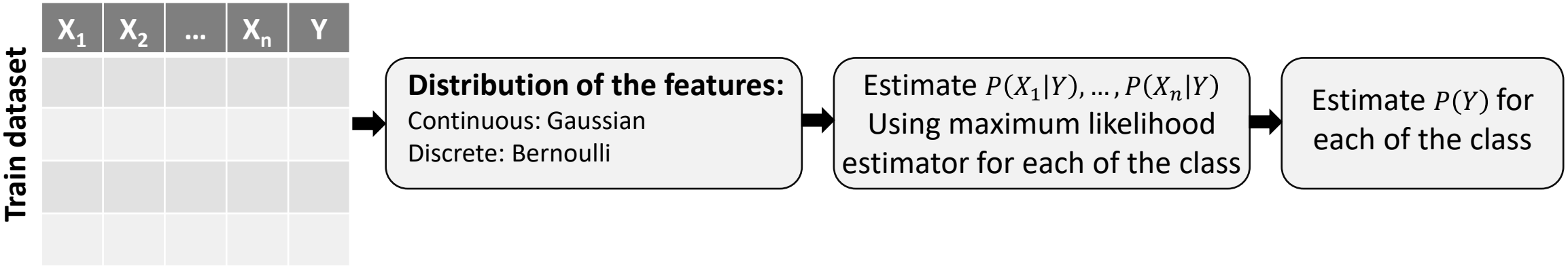
- Assumption: Conditional independence of likelihood function (Naïve assumption)

$$P(X_1, X_2, \ldots,X_n|Y) = P(X_1|Y) * P(X_2|Y) * \cdots * P(X_n|Y)$$

Train dataset

X_1	X_2	…	X_n	Y

Naïve Bayes classifier



- Estimated probabilities are used to predict posterior,

$$P(Y|X_1, X_2, \dots, X_n).$$

- Marginal, $P(X_1, X_2, \dots, X_n)$ is generally ignored in the calculation as both the classes have same marginal.
- The class with greater posterior probability is the predicted class.

Naïve Bayes classifier: example dataset1

- Let X_1 and X_2 be the features and Y ∈ [Red, Black] be the class to be predicted

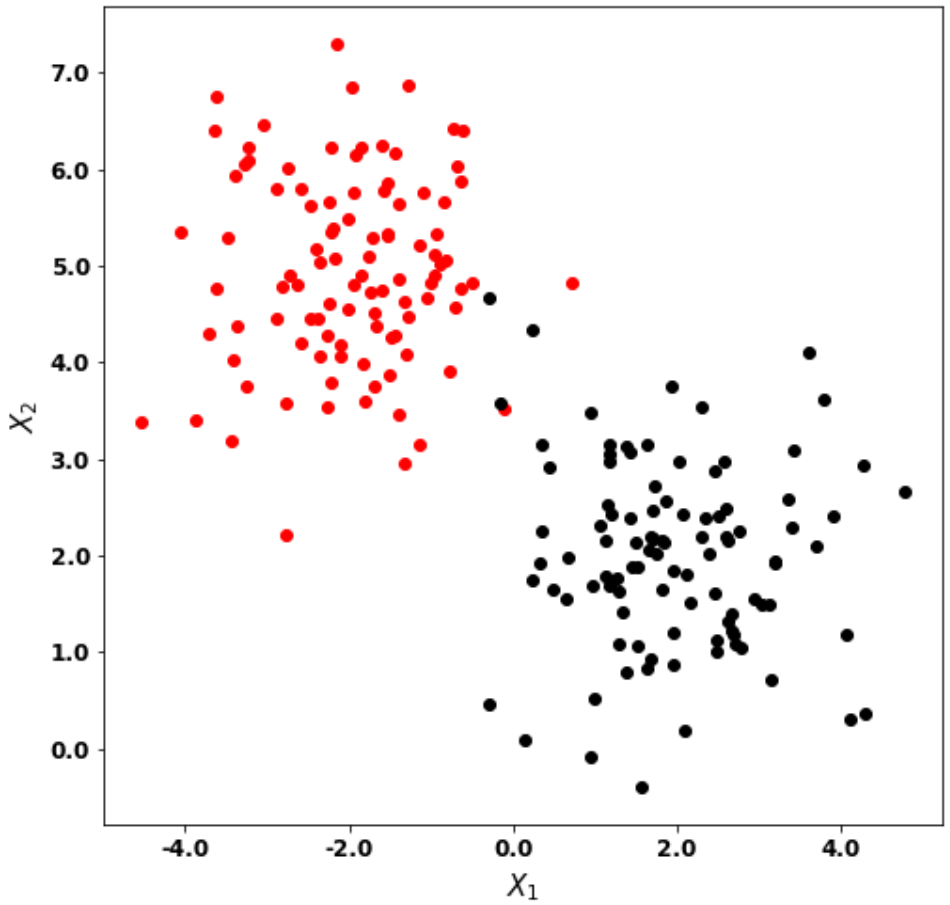


Figure 5: Scatter plot indicating the two classes

- To estimate $P(X_1|Y=Red)$ and $P(X_2|Y=Red)$

	Mean	Variance
$X_1\|Y=Red$	-1.9	0.98
$X_2\|Y=Red$	5.1	1.1

- To estimate $P(X_1|Y=Black)$ and $P(X_2|Y=Black)$

	Mean	Variance
$X_1\|Y=Black$	2.0	1.0
$X_2\|Y=Black$	1.9	1.1

- To estimate P(Y=Red) and P(Y=Black)

P(Y=Red)	0.5
P(Y=Black)	0.5

Naïve Bayes classifier: example dataset1

- Let X_1 and X_2 be the features and Y ∈ [Red, Black] be the class to be predicted

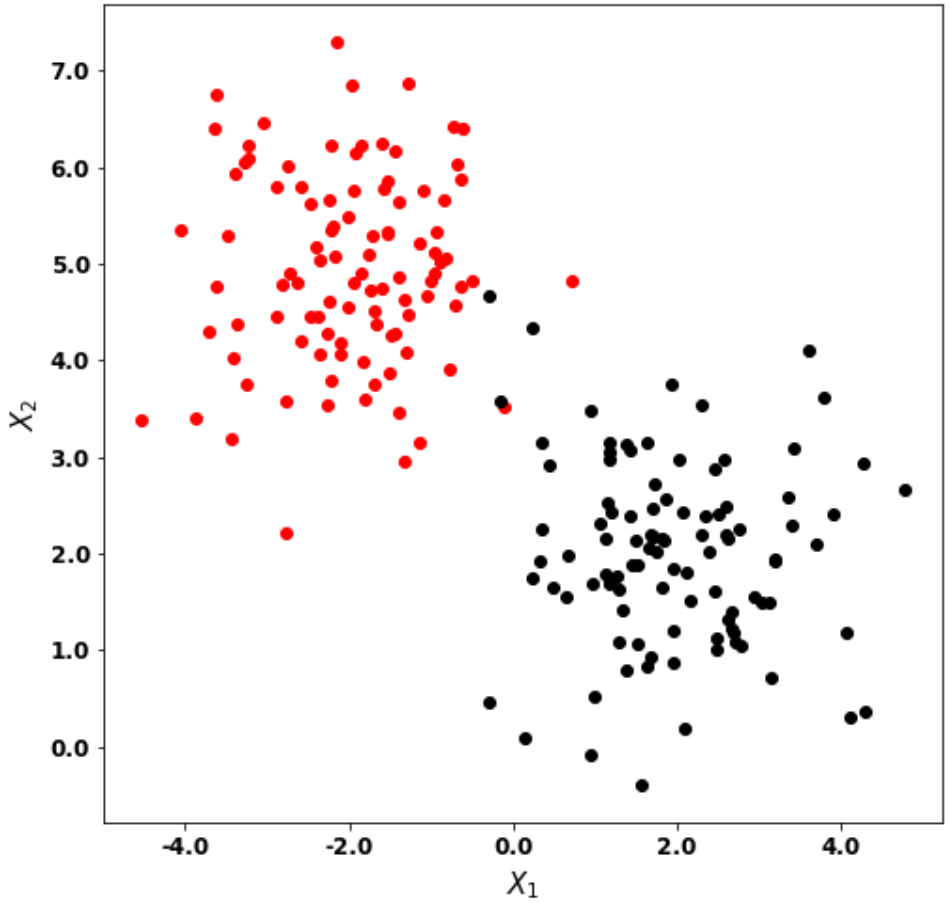


Figure 5: Scatter plot indicating the two classes

- For the class, Y = Red

$$P(X_1|Y=Red) = \frac{1}{\sqrt{2\pi(0.98)}} \exp\left(\frac{-1}{2}\left(\frac{X_1+1.9}{\sqrt{0.98}}\right)^2\right)$$

$$P(X_2|Y=Red) = \frac{1}{\sqrt{2\pi(1.1)}} \exp\left(\frac{-1}{2}\left(\frac{X_2-5.1}{\sqrt{1.1}}\right)^2\right)$$

$$P(Y=Red) = 0.5$$

- For the class, Y = Black

$$P(X_1|Y=Black) = \frac{1}{\sqrt{2\pi(1)}} \exp\left(\frac{-1}{2}\left(\frac{X_1-2}{\sqrt{1}}\right)^2\right)$$

$$P(X_2|Y=Black) = \frac{1}{\sqrt{2\pi(1.1)}} \exp\left(\frac{-1}{2}\left(\frac{X_2-1.9}{\sqrt{1.1}}\right)^2\right)$$

$$P(Y=Black) = 0.5$$

Obtained probabilities can be used to predict the probabilities, $P(Y=Red|X_1,X_2)$ and $P(Y=Black|X_1,X_2)$

Naïve Bayes classifier: example dataset2

- Let X_1 and X_2 be the features and Y ∈ [Red, Black] be the class to be predicted

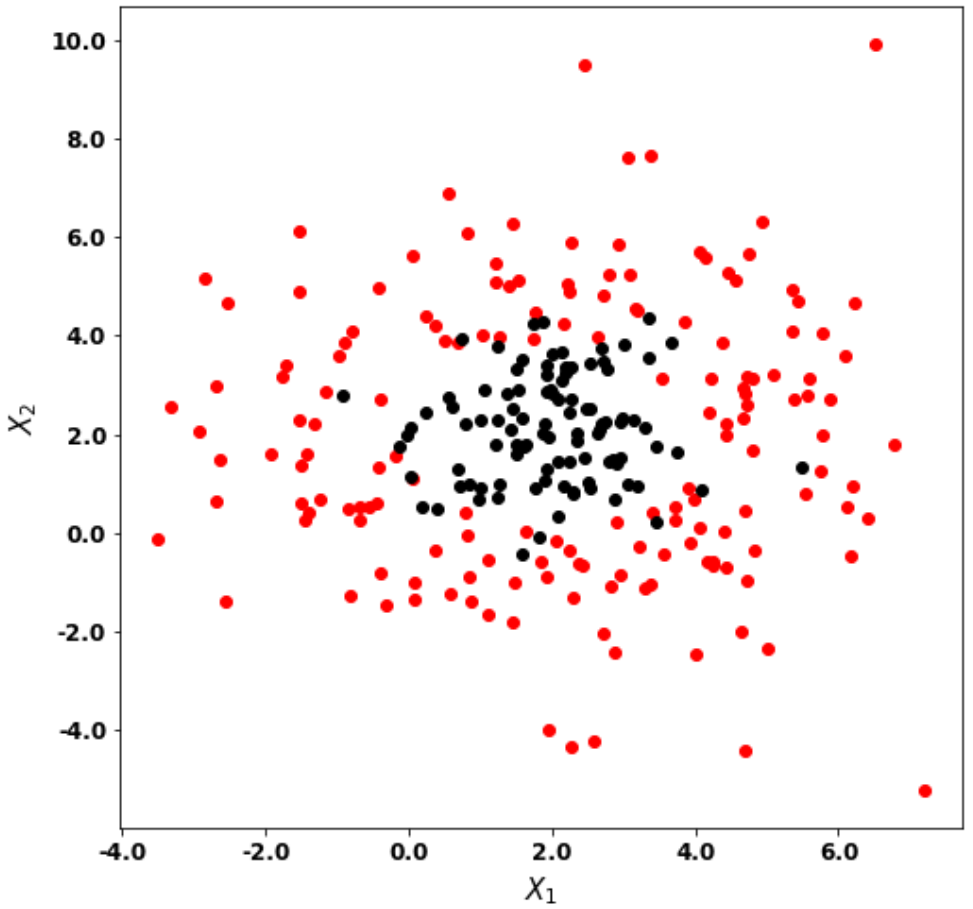


Figure 6: Scatter plot indicating the two classes

- To estimate $P(X_1|Y=Red)$ and $P(X_2|Y=Red)$

	Mean	Variance
$X_1\|Y=Red$	2.2	6.8
$X_2\|Y=Red$	1.9	8.0

- To estimate $P(X_1|Y=Black)$ and $P(X_2|Y=Black)$

	Mean	Variance
$X_1\|Y=Black$	2.0	1.0
$X_2\|Y=Black$	1.9	1.1

- To estimate P(Y=Red) and P(Y=Black)

P(Y=Red)	0.5
P(Y=Black)	0.5

Naïve Bayes classifier: example dataset2

- Let X_1 and X_2 be the features and Y ∈ [Red, Black] be the class to be predicted

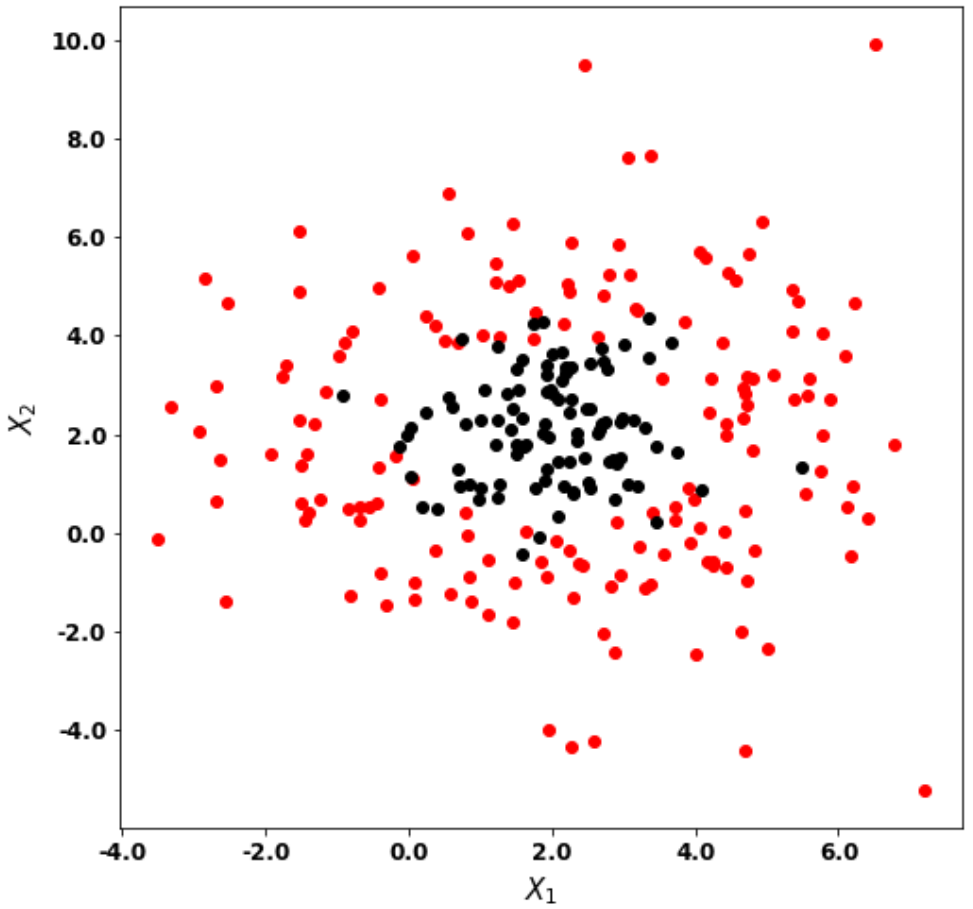


Figure 6: Scatter plot indicating the two classes

- For the class, Y = Red

$$P(X_1|Y=Red) = \frac{1}{\sqrt{2\pi(6.8)}} \exp\left(\frac{-1}{2}\left(\frac{X_1-2.2}{\sqrt{6.8}}\right)^2\right)$$

$$P(X_2|Y=Red) = \frac{1}{\sqrt{2\pi(8.0)}} \exp\left(\frac{-1}{2}\left(\frac{X_2-1.9}{\sqrt{8.0}}\right)^2\right)$$

$$P(Y=Red) = 0.5$$

- For the class, Y = Black

$$P(X_1|Y=Black) = \frac{1}{\sqrt{2\pi(1)}} \exp\left(\frac{-1}{2}\left(\frac{X_1-2}{\sqrt{1}}\right)^2\right)$$

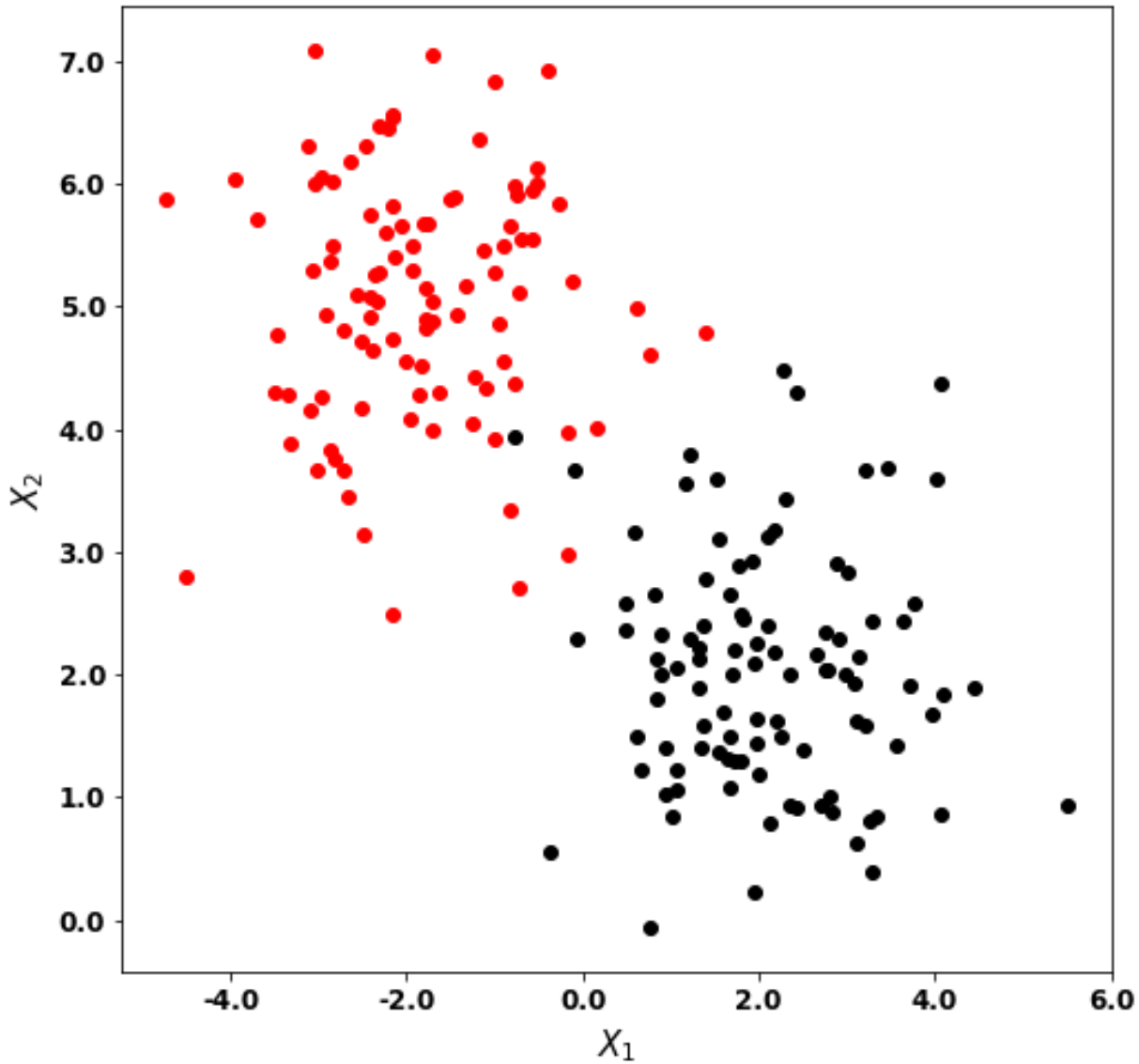
$$P(X_2|Y=Black) = \frac{1}{\sqrt{2\pi(1.1)}} \exp\left(\frac{-1}{2}\left(\frac{X_2-1.9}{\sqrt{1.1}}\right)^2\right)$$

$$P(Y=Black) = 0.5$$

Obtained probabilities can be used to predict the probabilities, $P(Y=Red|X_1,X_2)$ and $P(Y=Black|X_1,X_2)$

Comparing naïve Bayes classifier and Logistic regression

Example dataset1



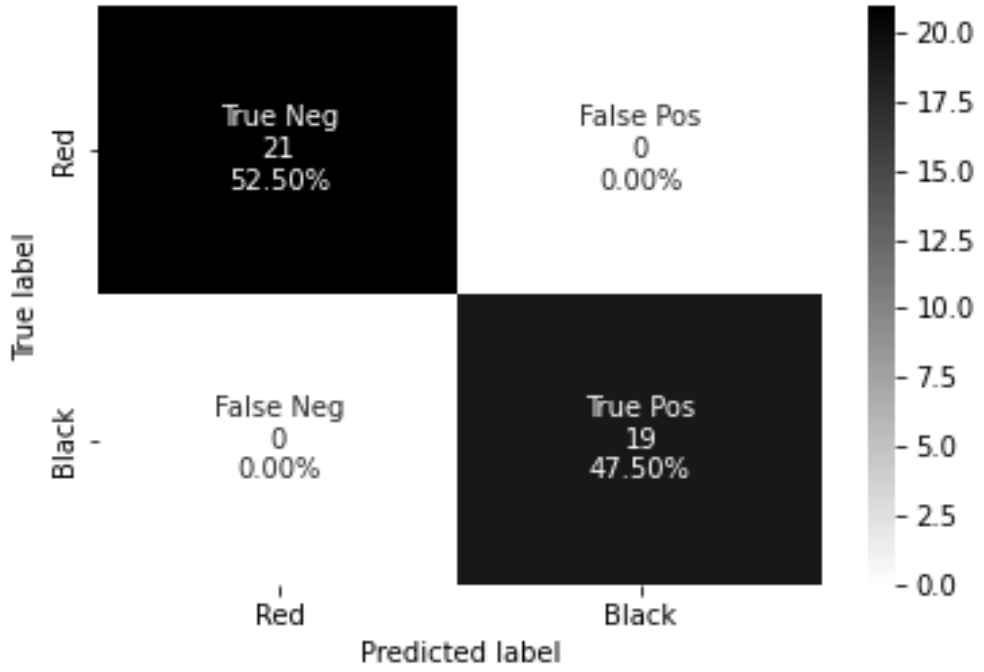
Example dataset1

Logistic regression

Accuracy	1
Precision	1
Recall	1
F1 score	1

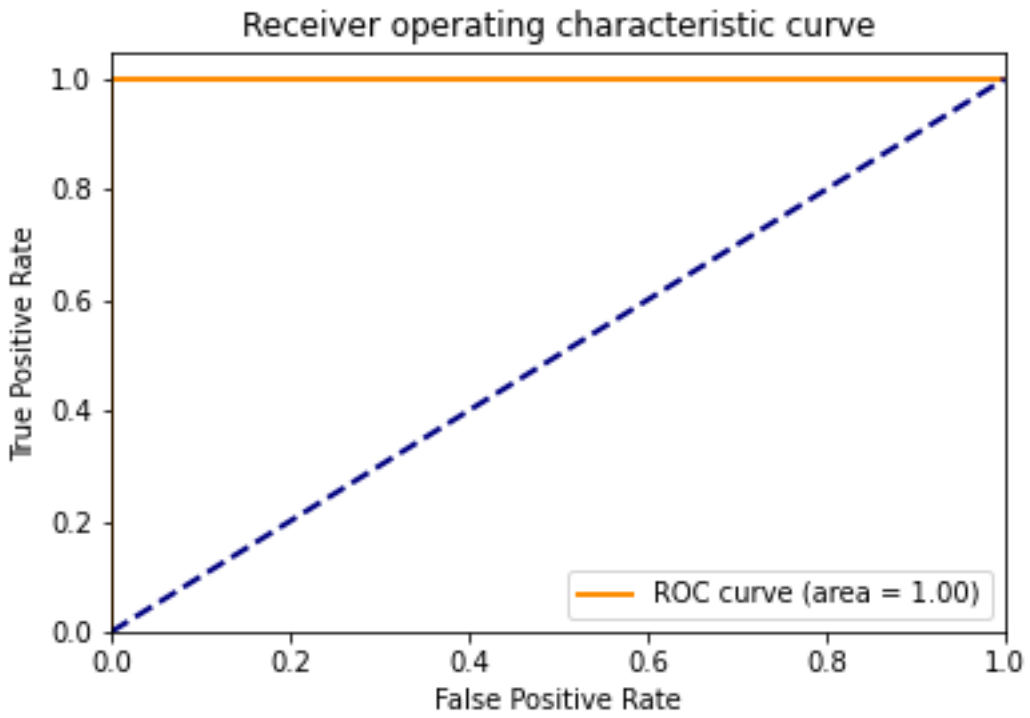
Gaussian naïve Bayes

Accuracy	1
Precision	1
Recall	1
F1 score	1

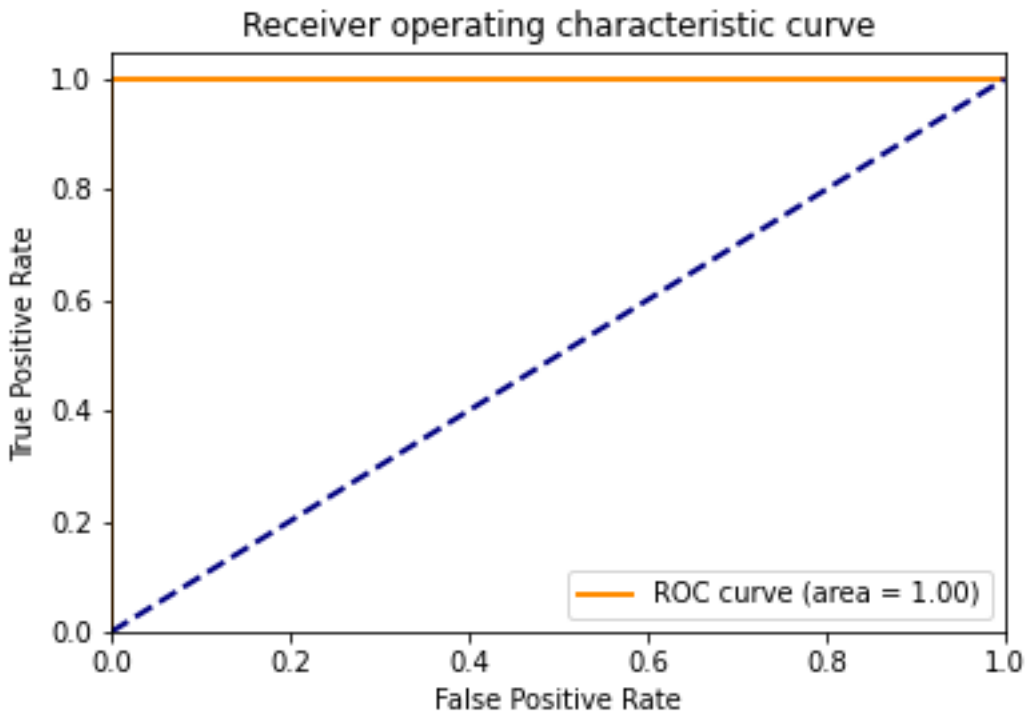


Example dataset1

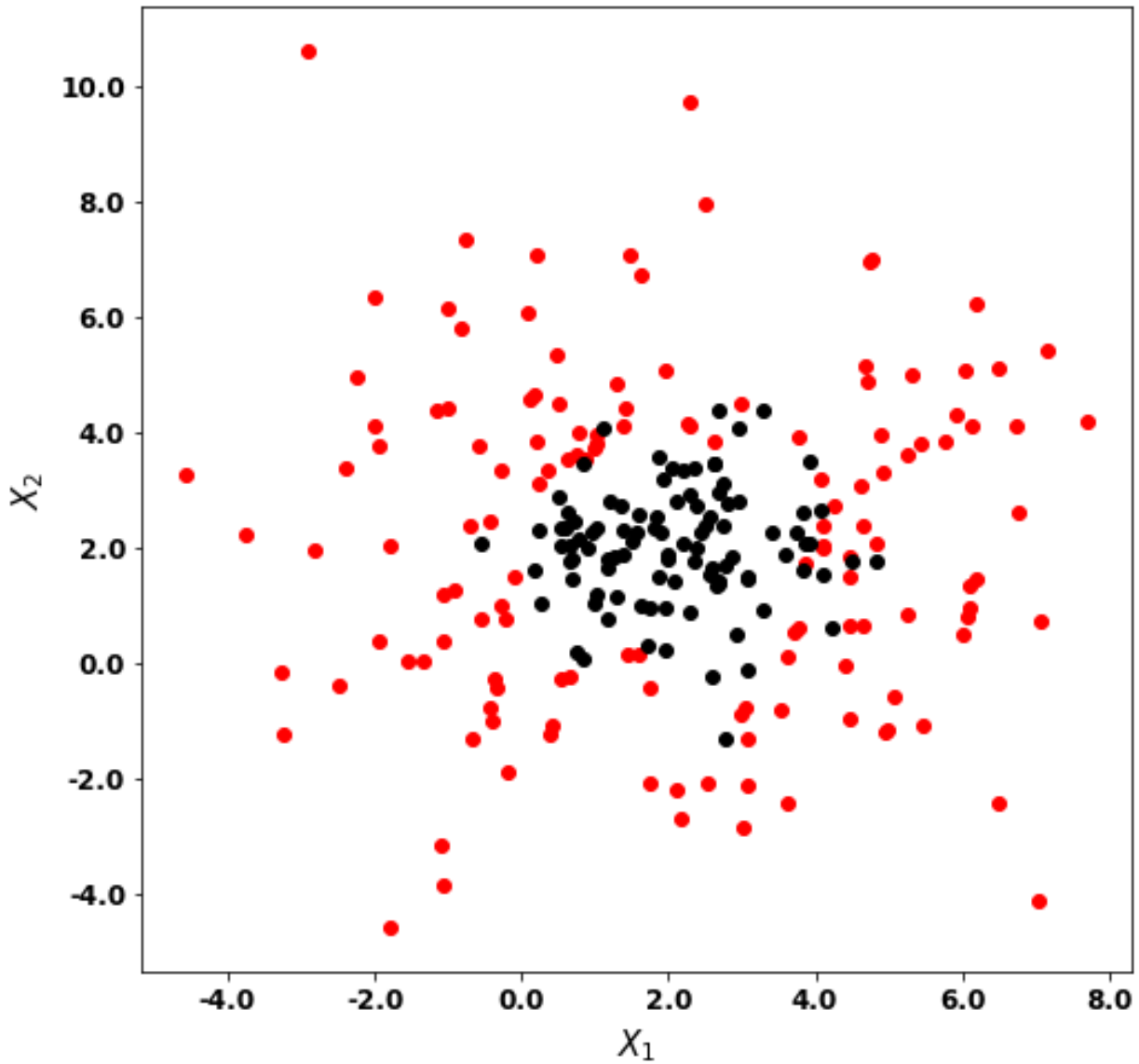
Logistic regression



Gaussian naïve Bayes



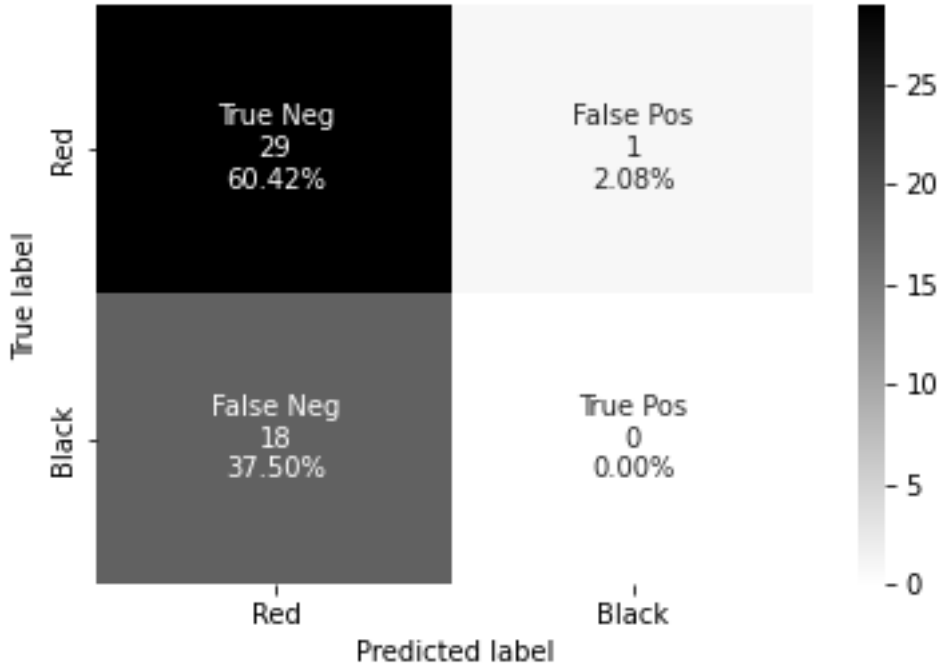
Example dataset2



Example dataset2

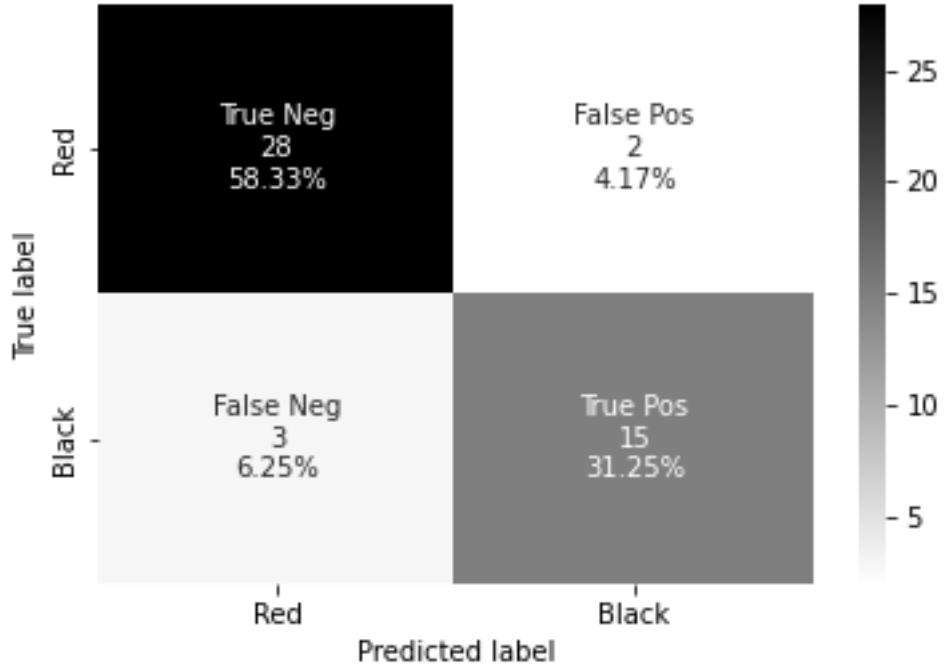
Logistic regression

Accuracy	0.6
Precision	0
Recall	0
F1 score	NaN



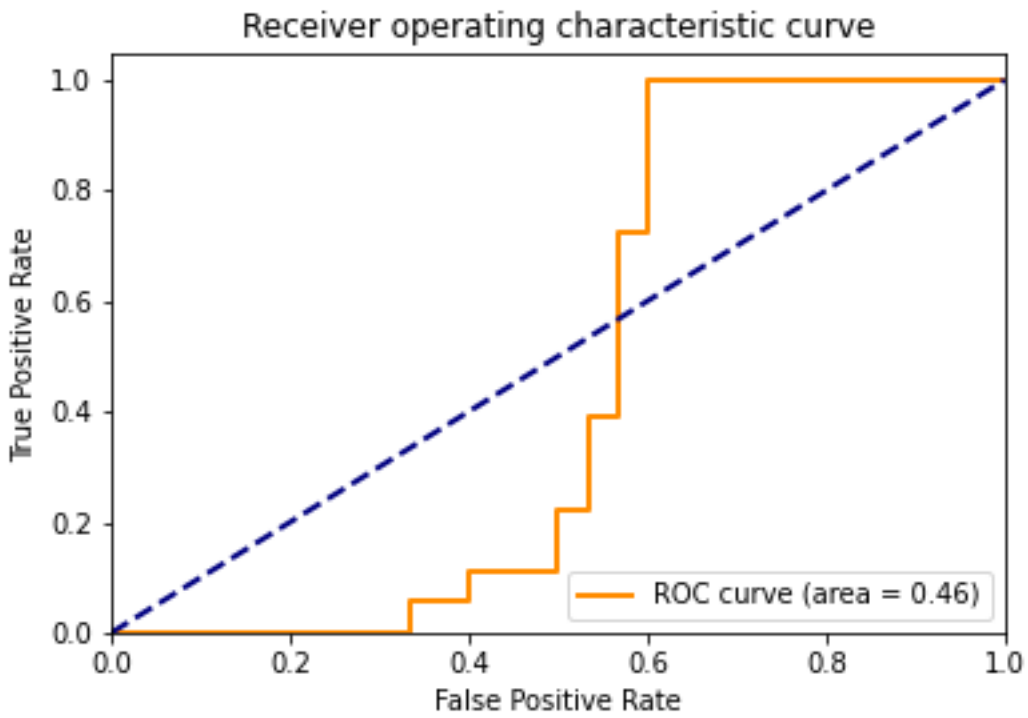
Gaussian naïve Bayes

Accuracy	0.9
Precision	0.88
Recall	0.83
F1 score	0.86

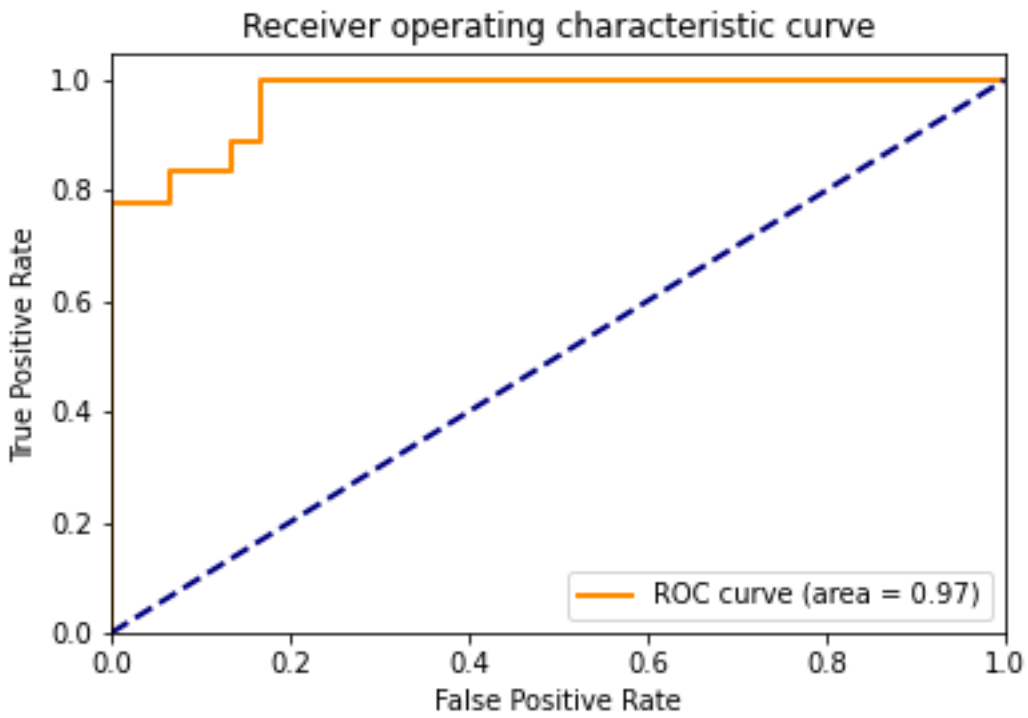


Example dataset2

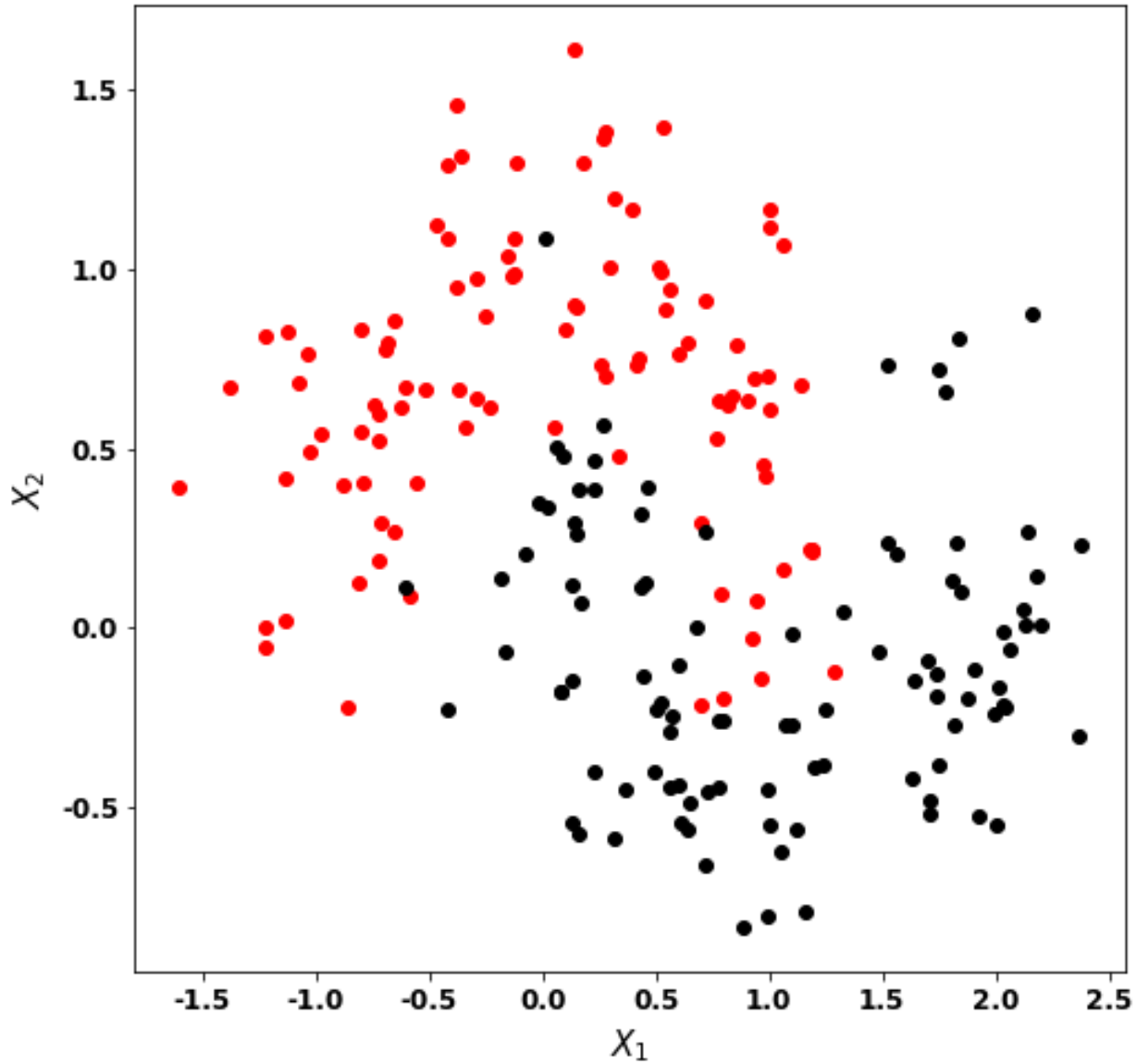
Logistic regression



Gaussian naïve Bayes



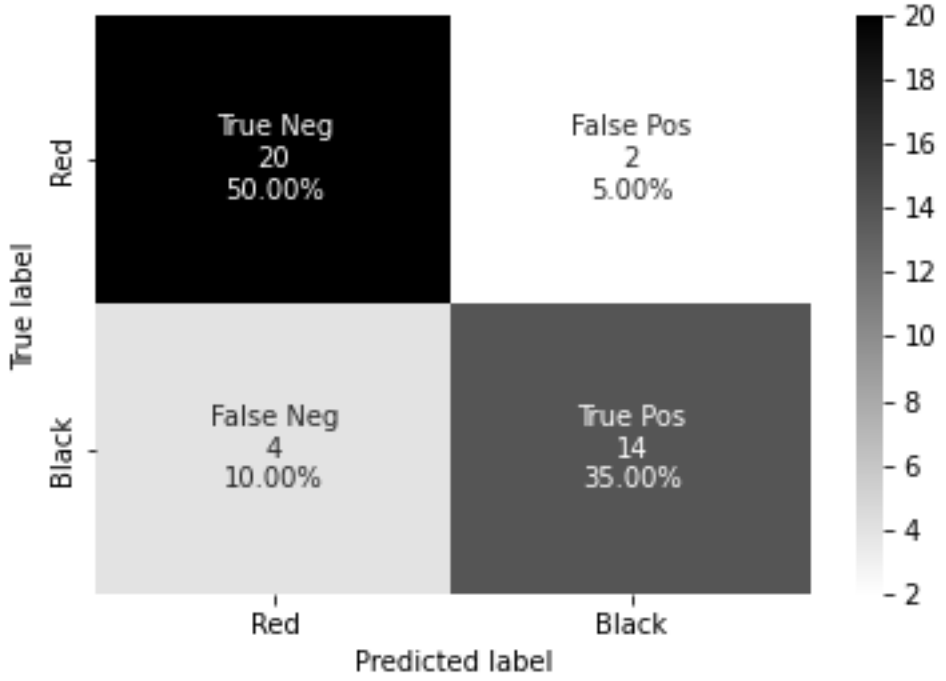
Example dataset3



Example dataset3

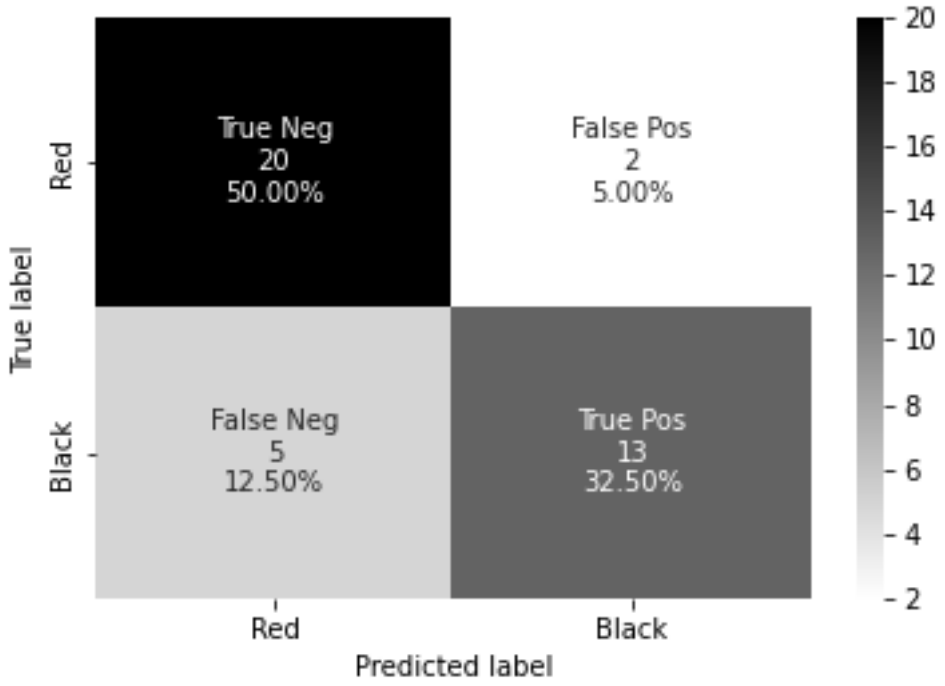
Logistic regression

Accuracy	0.85
Precision	0.88
Recall	0.78
F1 score	0.82



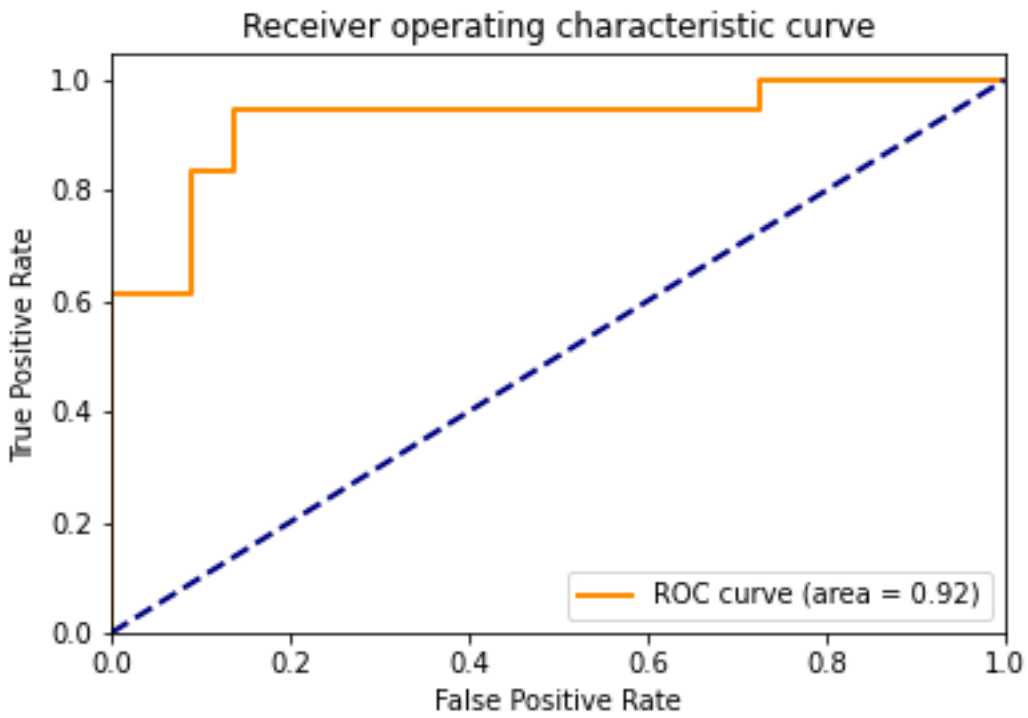
Gaussian naïve Bayes

Accuracy	0.82
Precision	0.87
Recall	0.72
F1 score	0.79

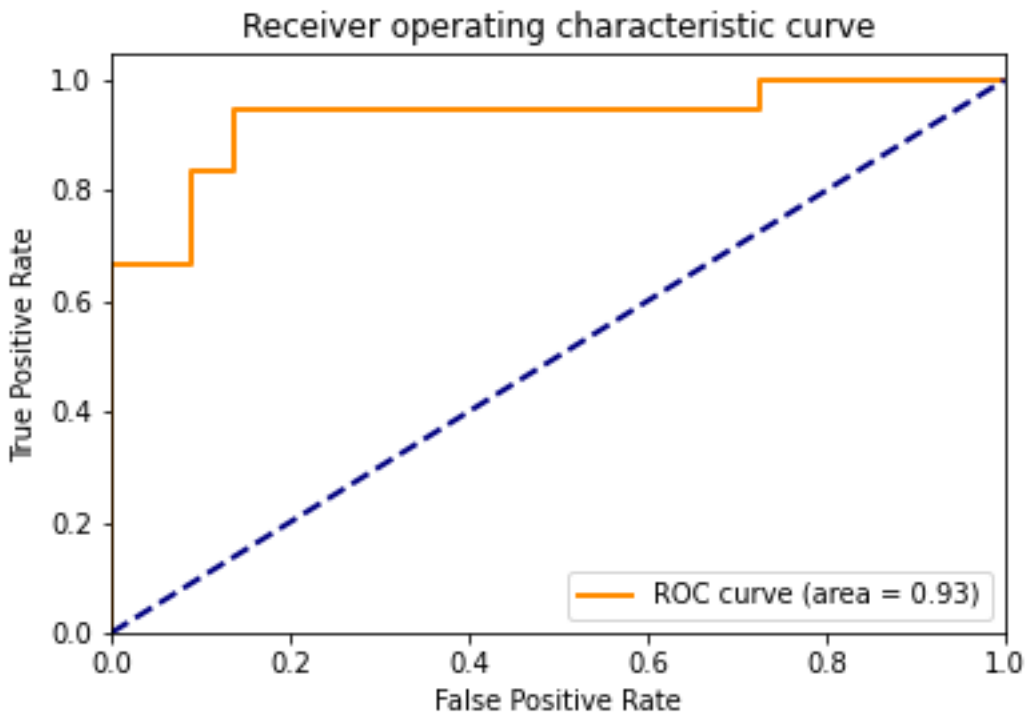


Example dataset3

Logistic regression



Gaussian naïve Bayes

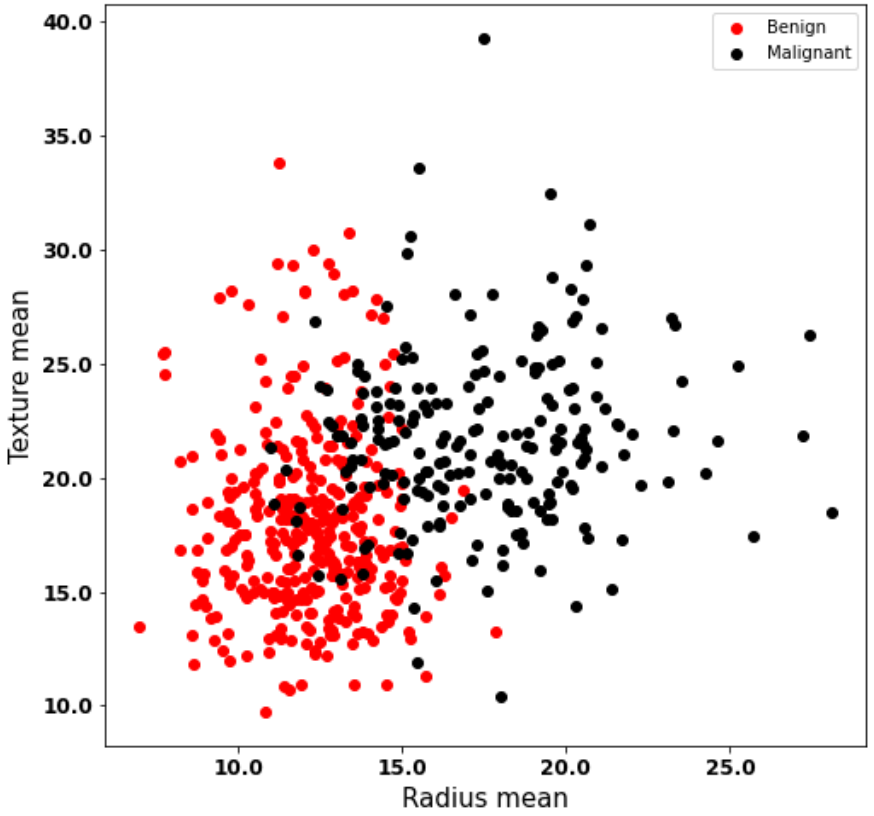


Cancer geometry data

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

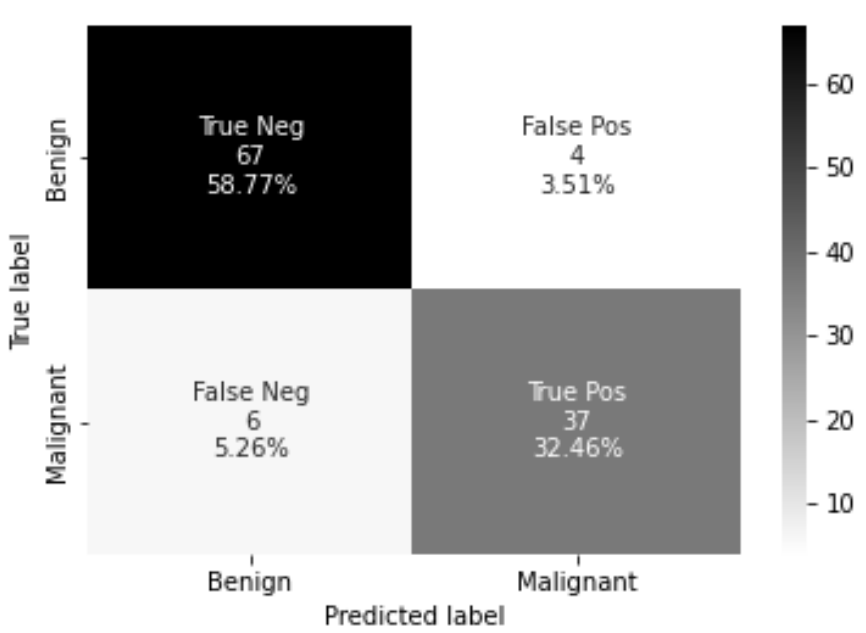
Features used for classification: radius mean and texture mean



Cancer geometry data

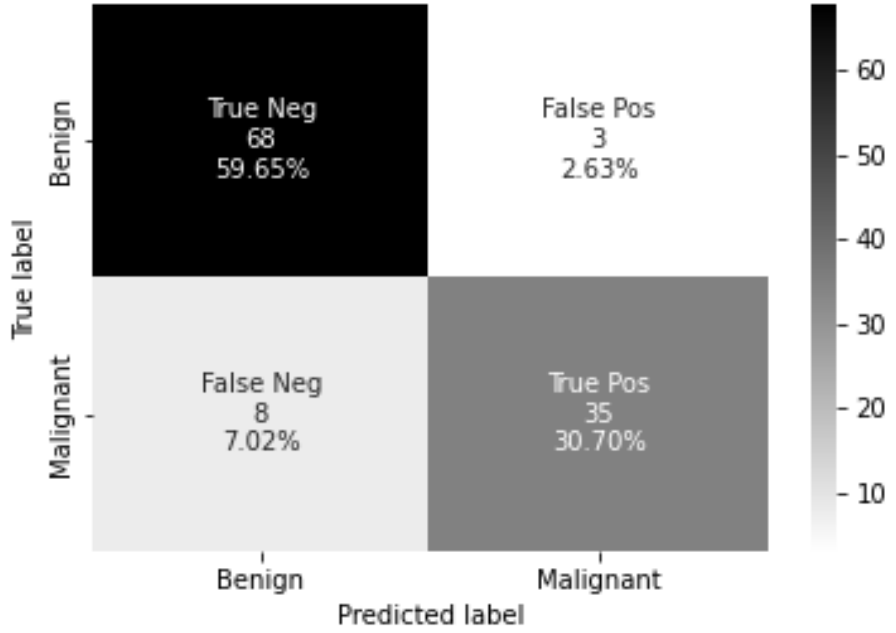
Logistic regression

Accuracy	0.91
Precision	0.90
Recall	0.86
F1 score	0.88



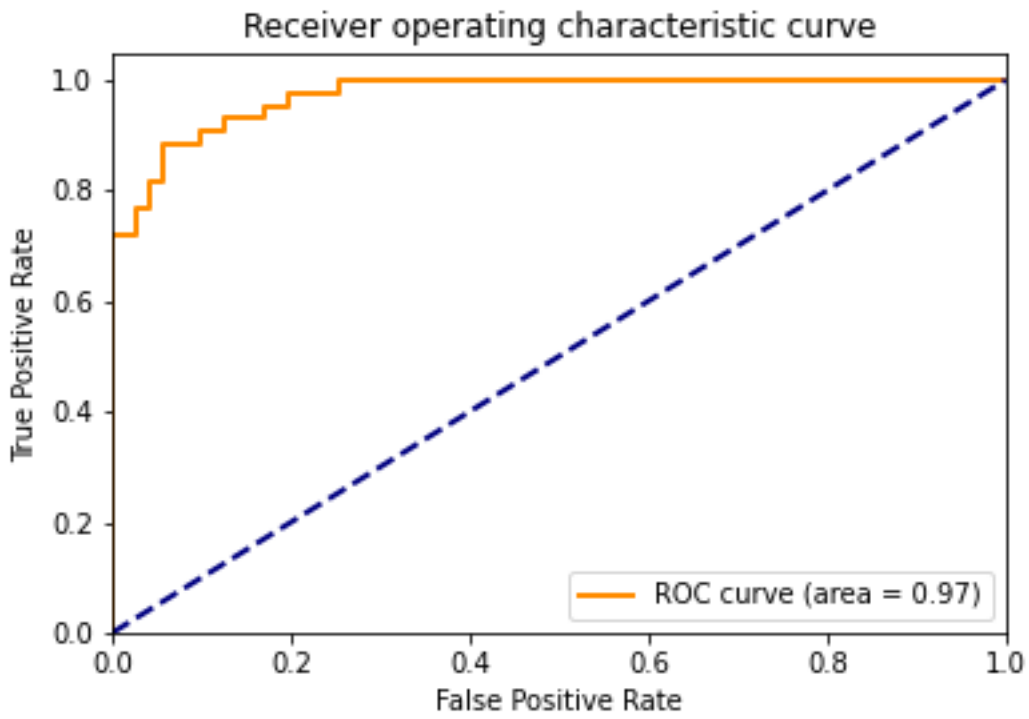
Gaussian naïve Bayes

Accuracy	0.9
Precision	0.92
Recall	0.81
F1 score	0.86

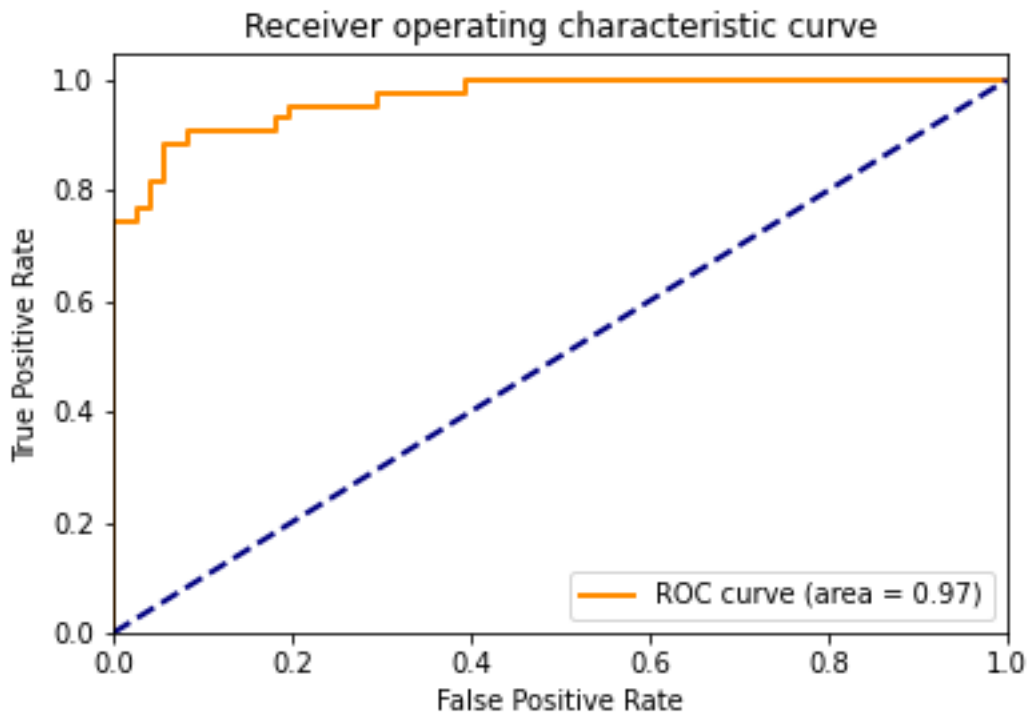


Cancer geometry data

Logistic regression

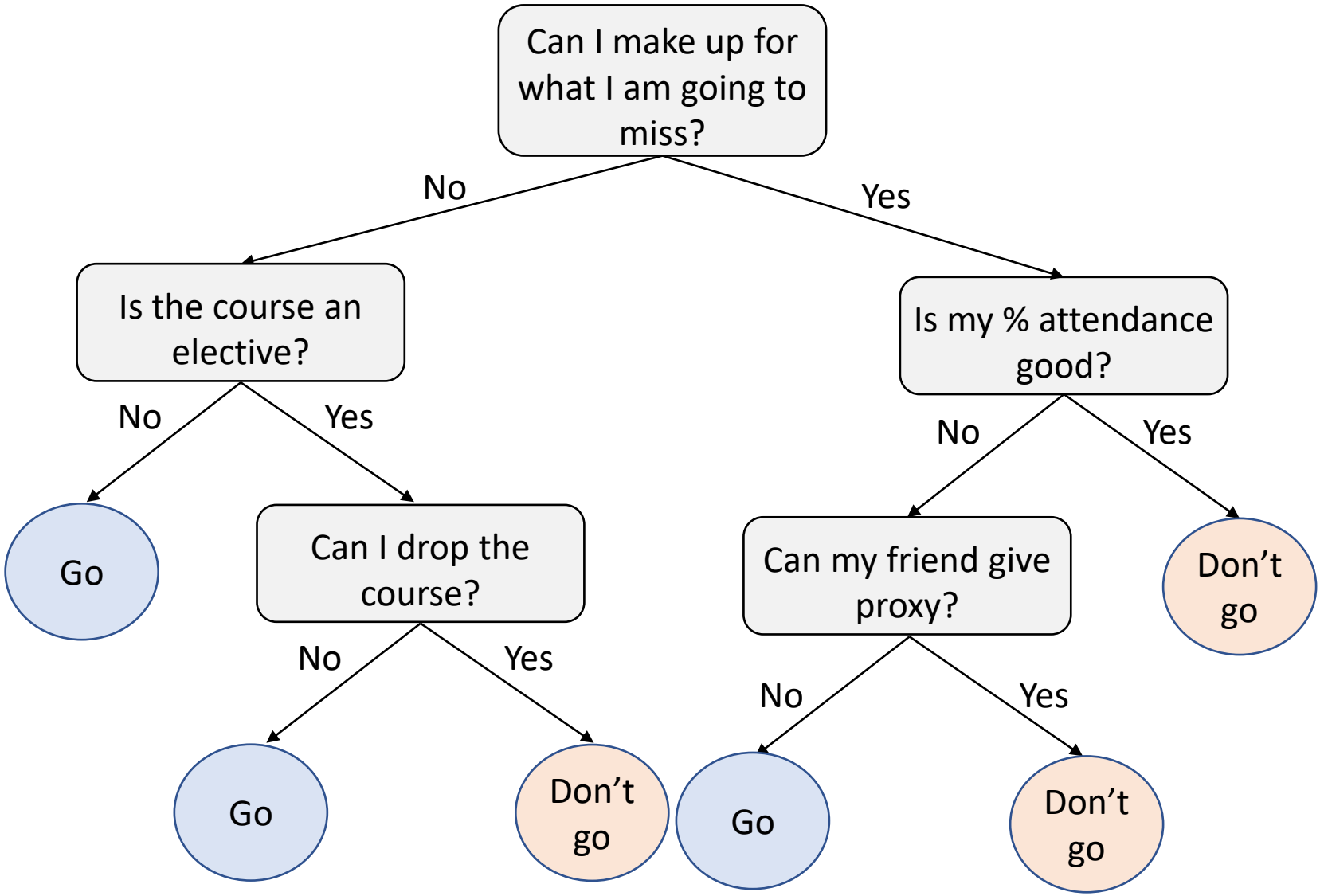


Gaussian naïve Bayes

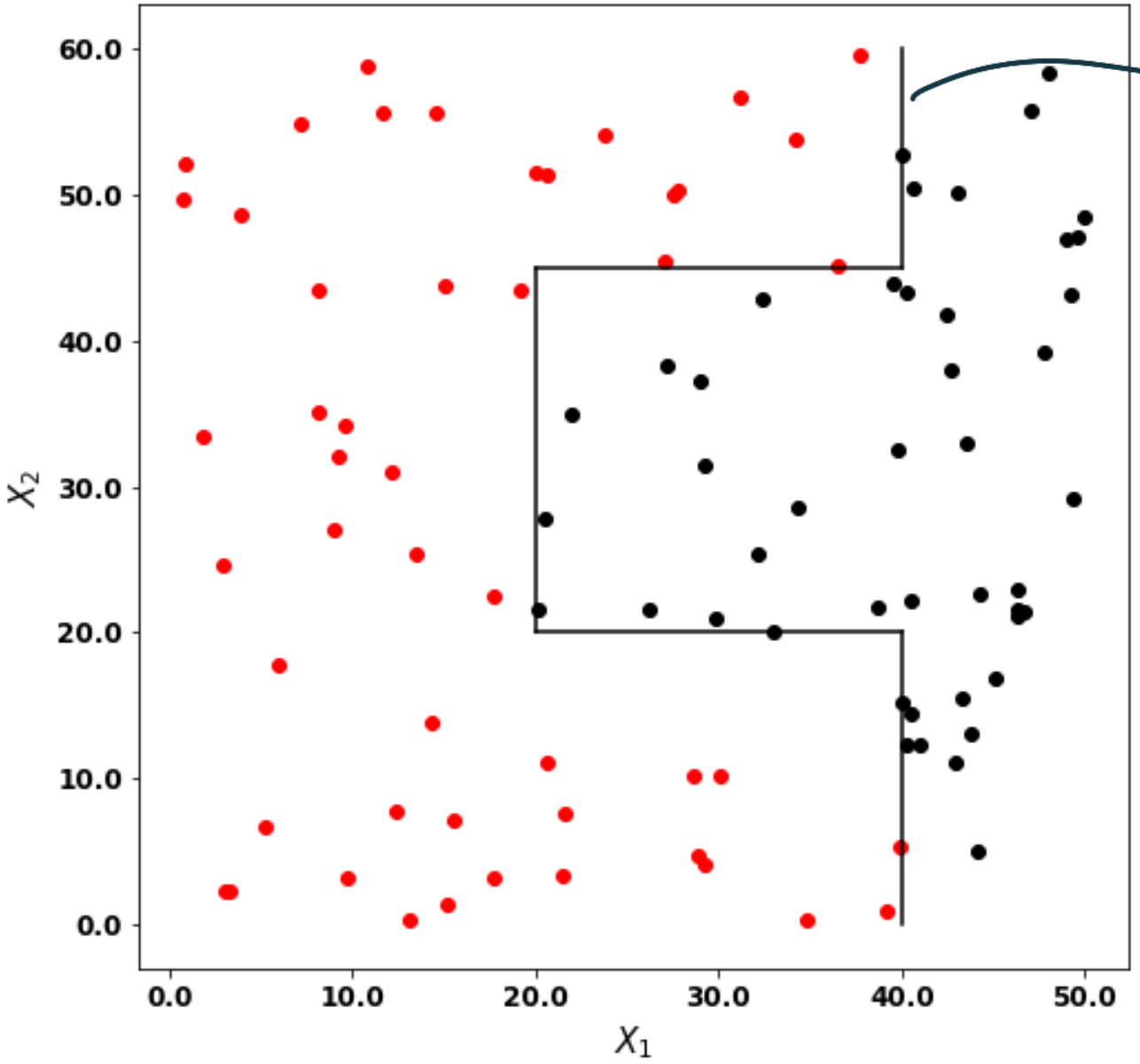


Decision Tree Classification

Should I need to go to class today?

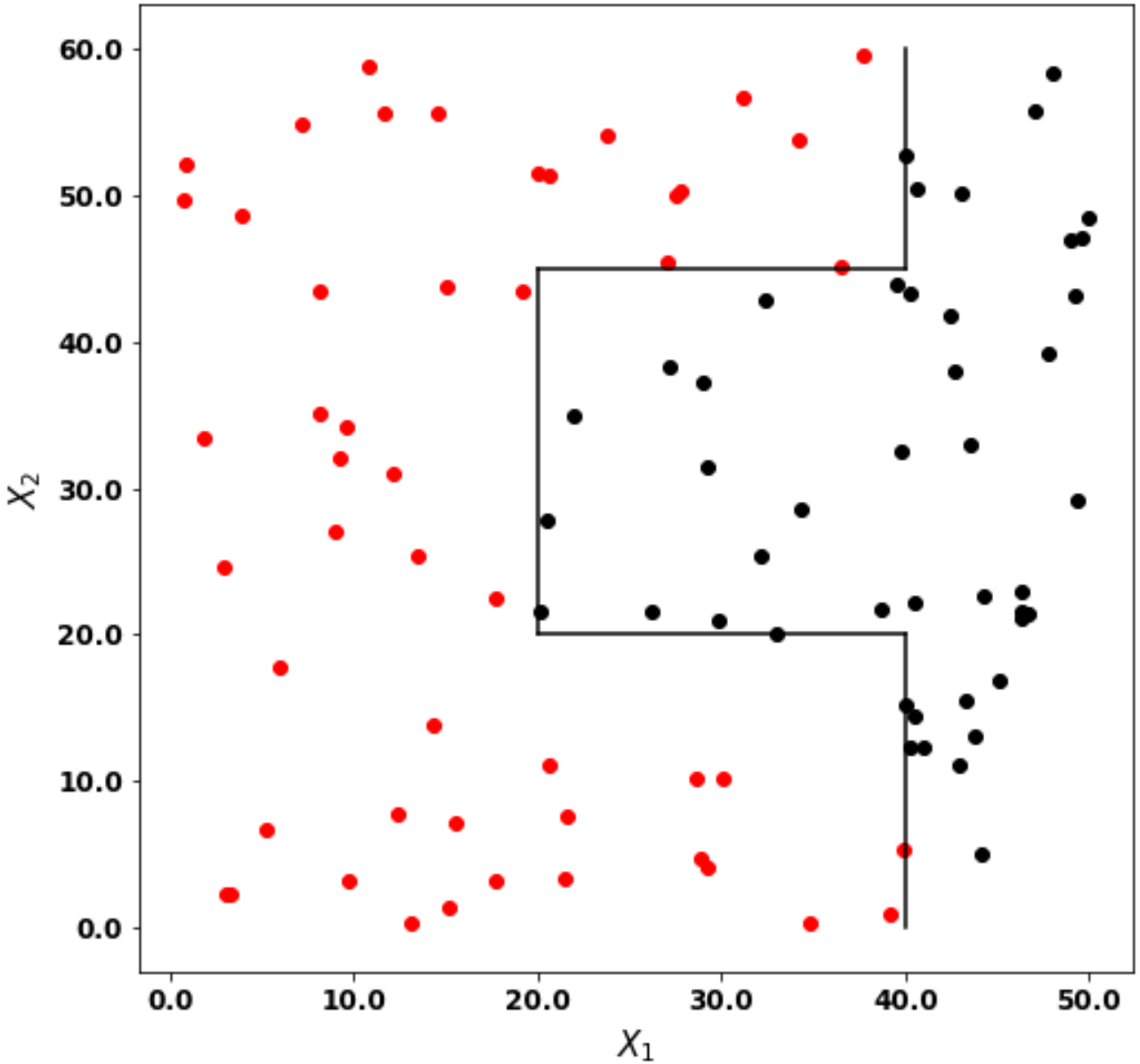


Red or black?

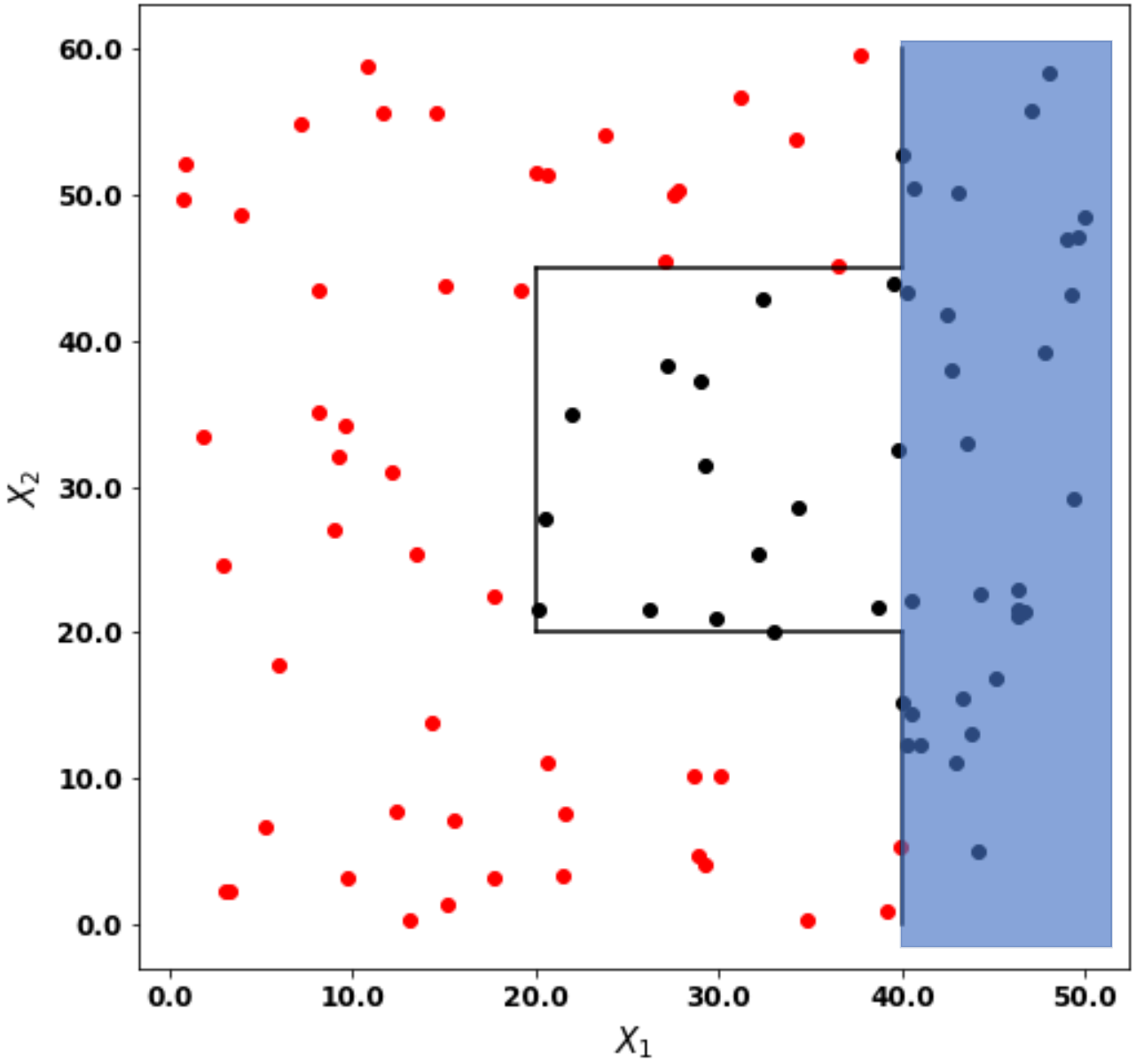


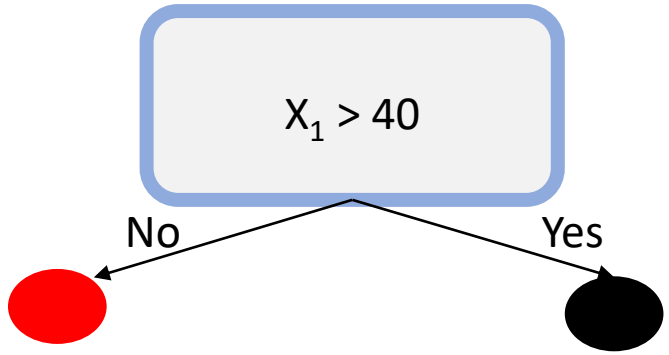
How do we create this boundary?

Red or black?

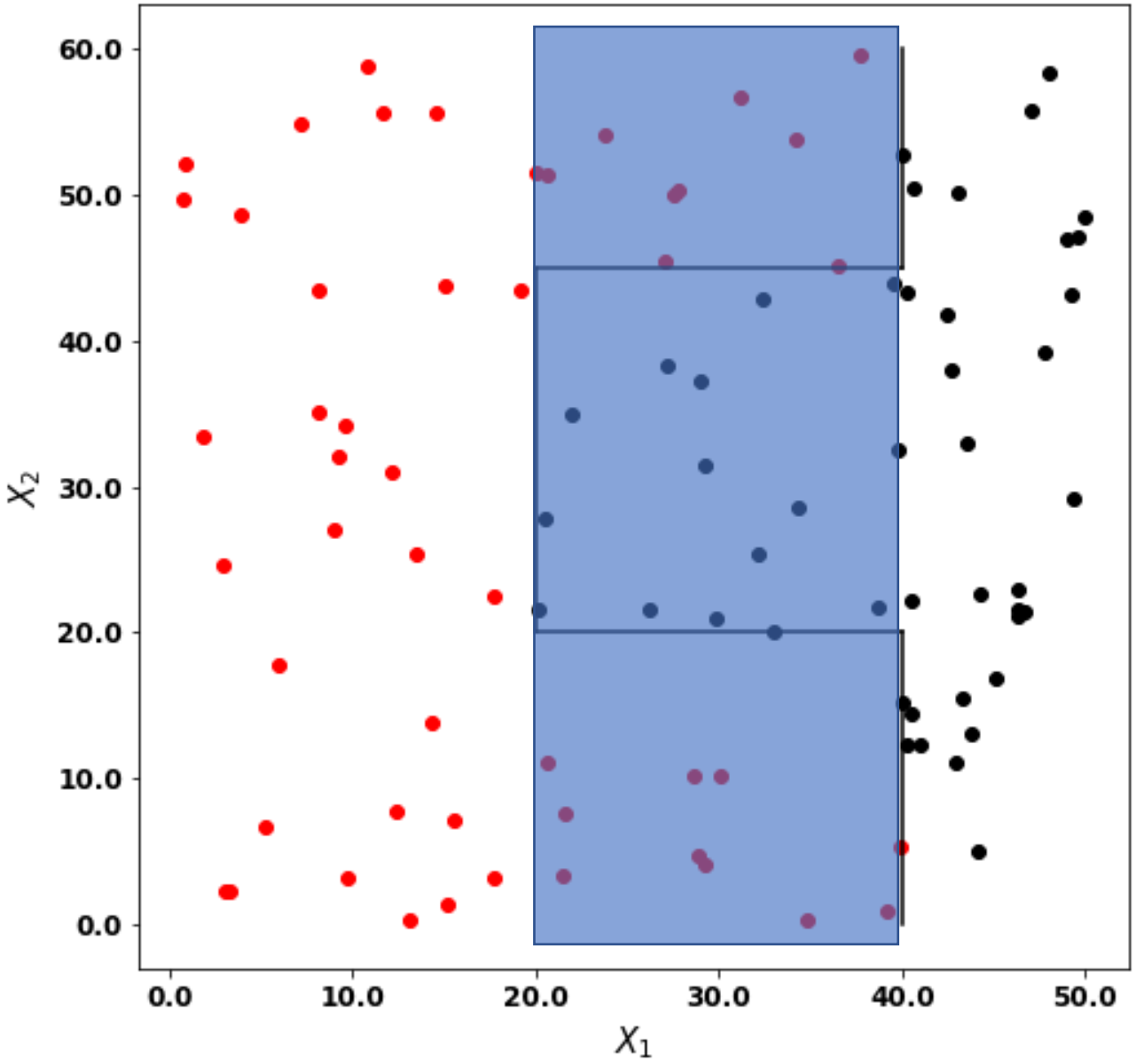


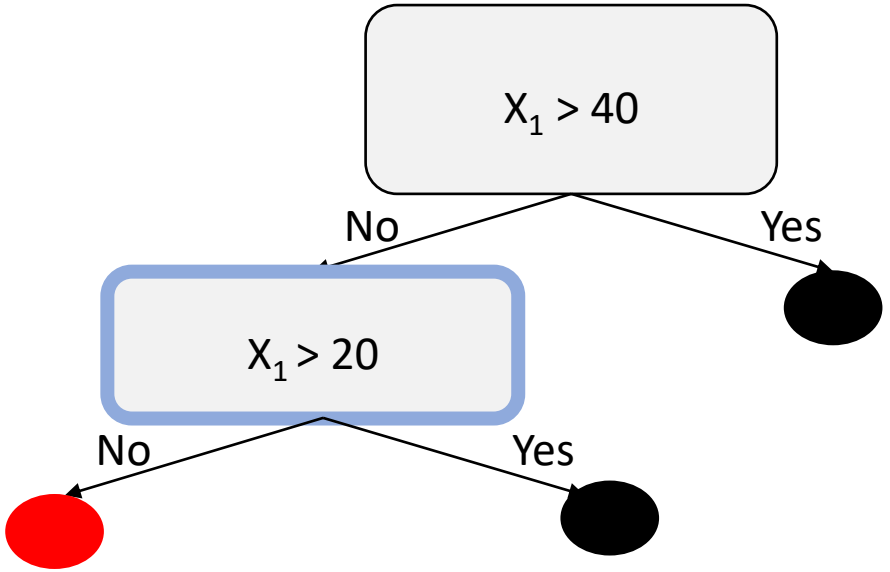
Red or black?


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

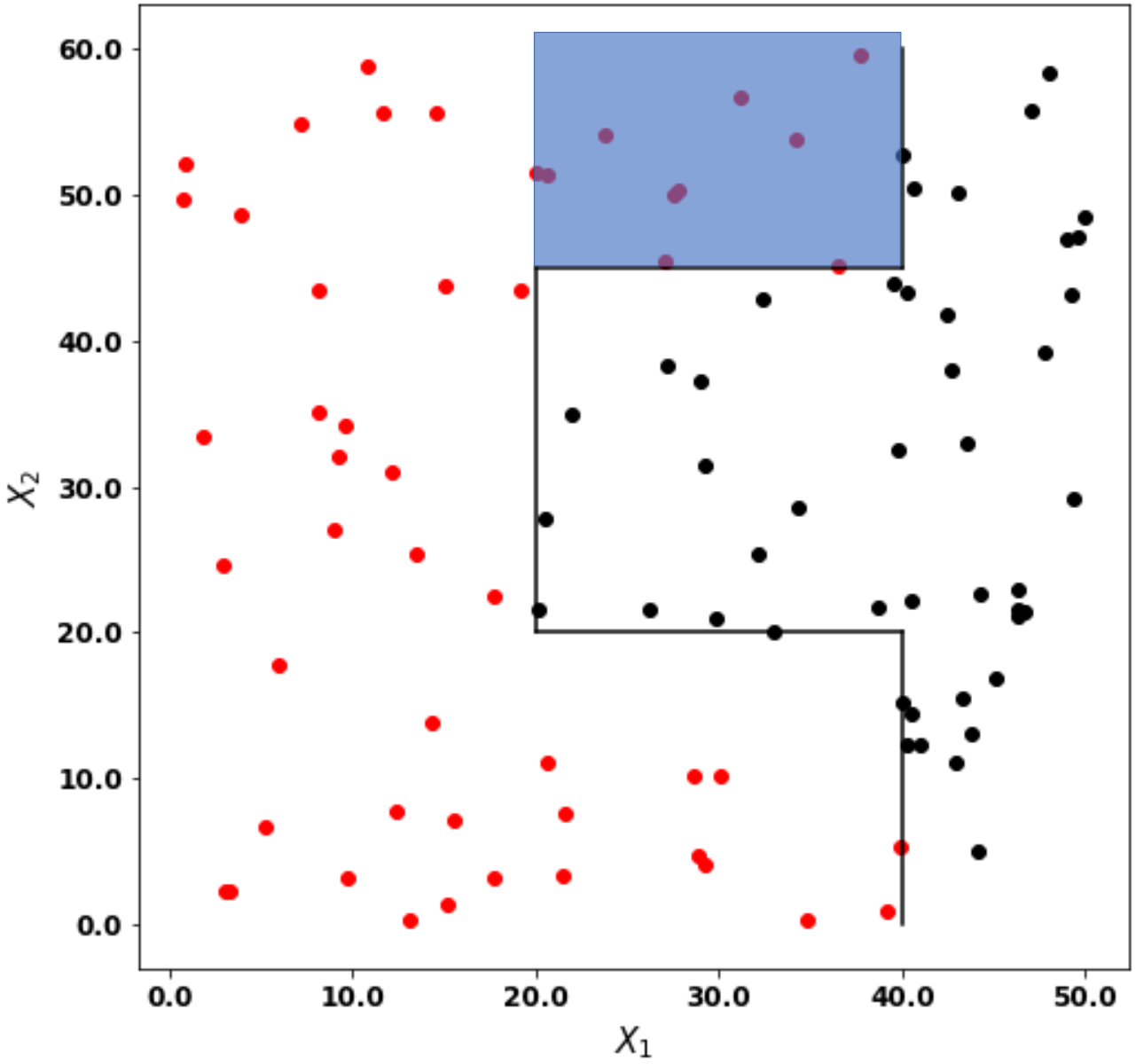

$X_1 > 40$
No
Yes

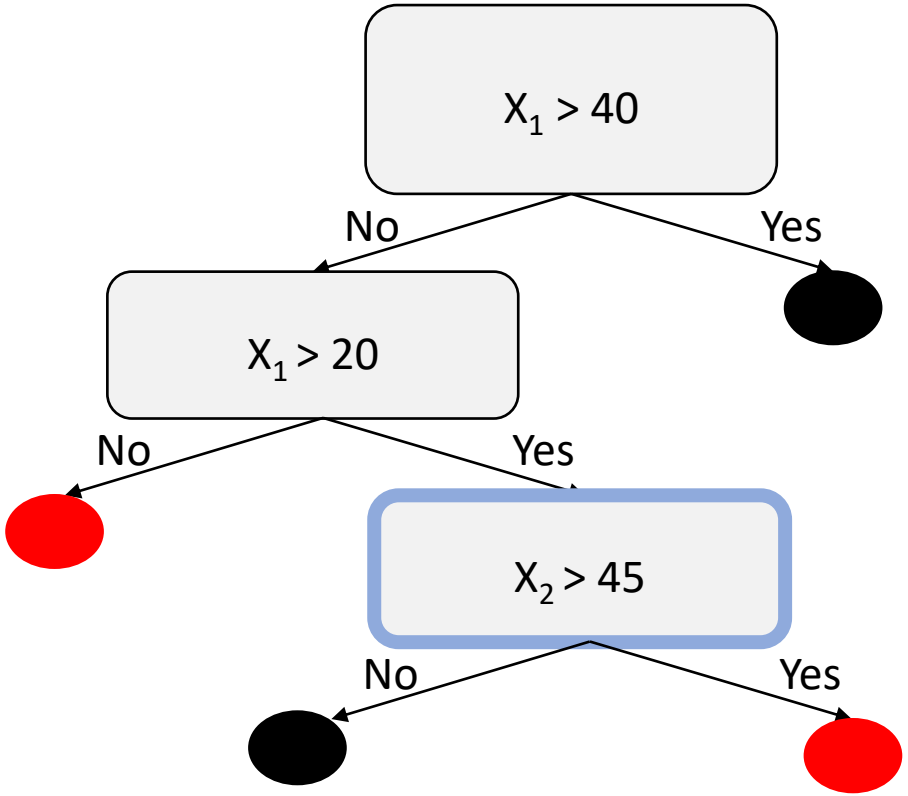
Red or black?


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

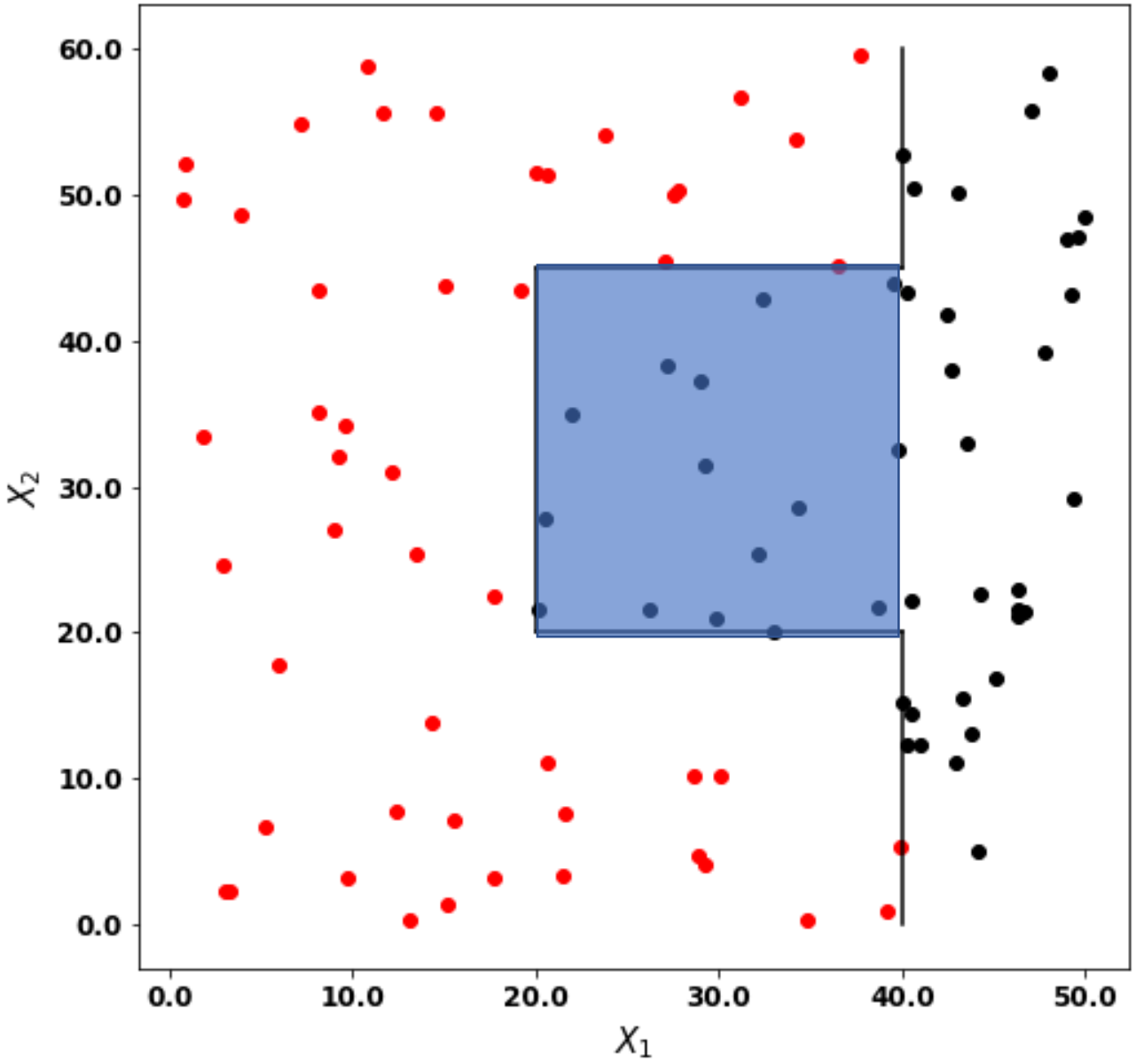

$X_1 > 40$
No
Yes
$X_1 > 20$
No
Yes

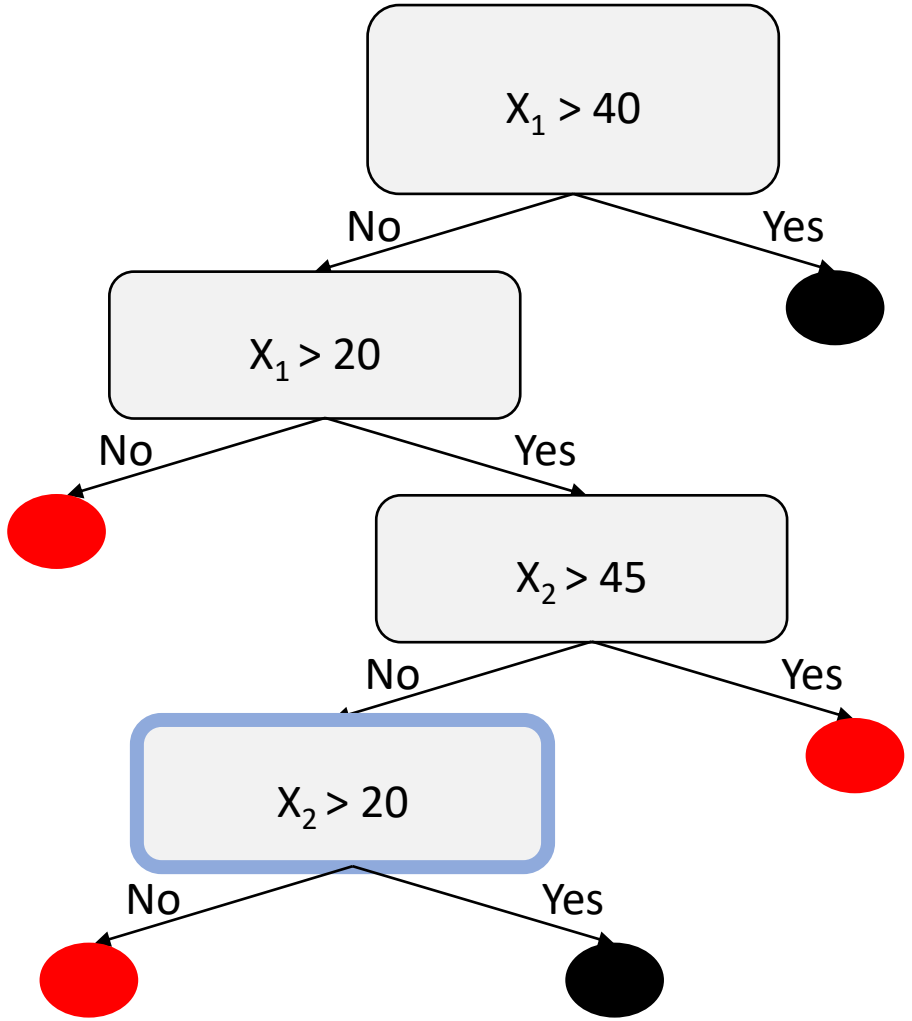
Red or black?


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

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

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 the most abundant training observations.

Decision trees for classification task

Decision tree algorithm:

- Begin with a 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.

Decision trees for classification task

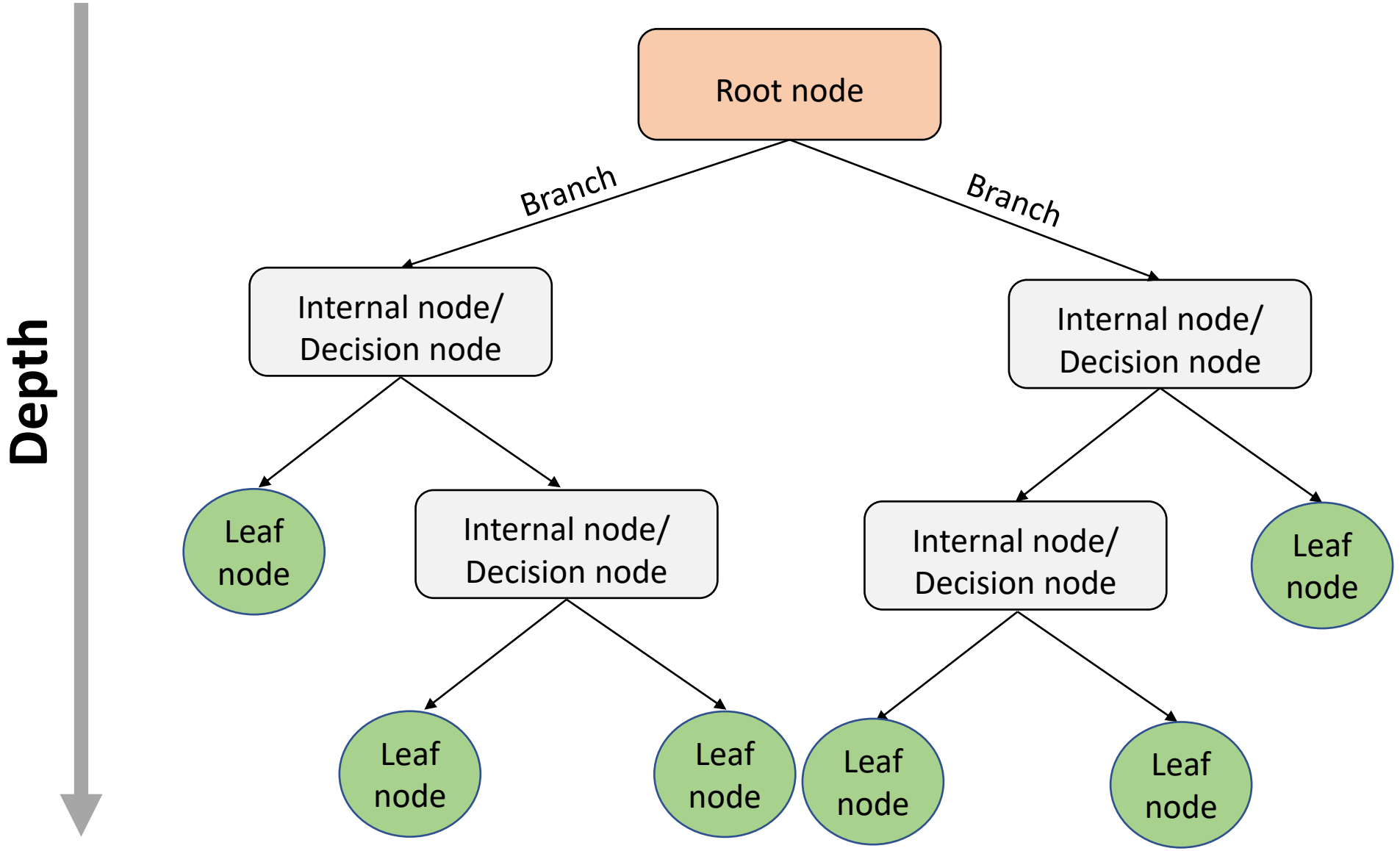
Advantages:

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

Disadvantages:

- Not robust to change in dataset.

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

How to choose optimal split?

Classification error rate

$$1 - \max_k(\hat{p}_{mk})$$

$\hat{p}_{mk}$ → Proportion of the kth class in training dataset in the mth region.

Classification Error rate:
Not sensitive with tree-growing.

How to choose optimal split?

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

Binary classification

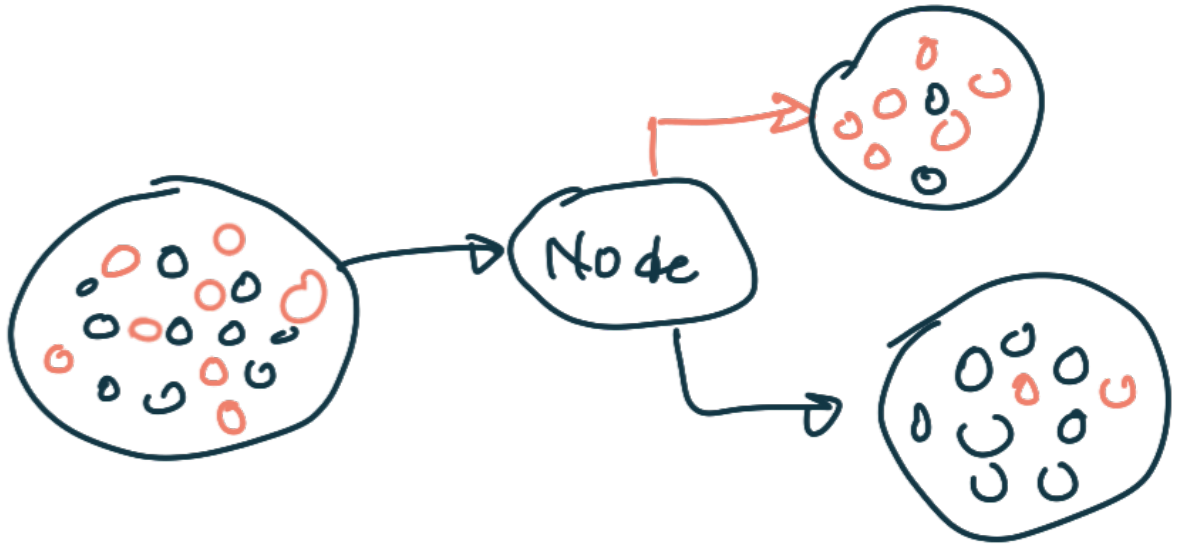
Gini Index : Node Purity.

How to choose optimal split?

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 a 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

Tree Pruning

Tree : Good on Training data

High depth tree → over fitting

→ Poor test error

Tree Pruning

Use greedy algorithm:

- Grow Biggest tree possible.
- Start pruning by merging leaf nodes and intermediate nodes such that test performance improves.

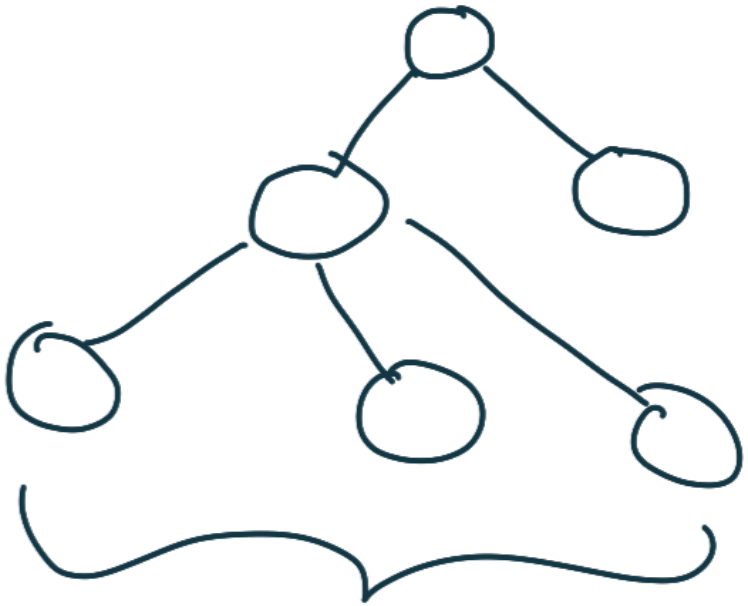
* cost-complexity Pruning
  - concept of Regularization

$$\min \underbrace{R(T)}_{\text{error}} + \alpha \underbrace{|T|}_{\text{number of leaf nodes}}$$

Large $\alpha \rightarrow$ Small Tree

small $\alpha \rightarrow$ Large Tree

Multiway Split vs Binary Split



Multiway split

Handle with care

→ sparsity

- Hard to interpret

→ Favors attributes with more values

* Use information gain

Cost-Associated Miss Classification

$L_{KK'}$ → Loss function

How to account for it?

$$\sum_{K=1}^{K} \hat{p}_{mK}(1 - \hat{p}_{mK})$$

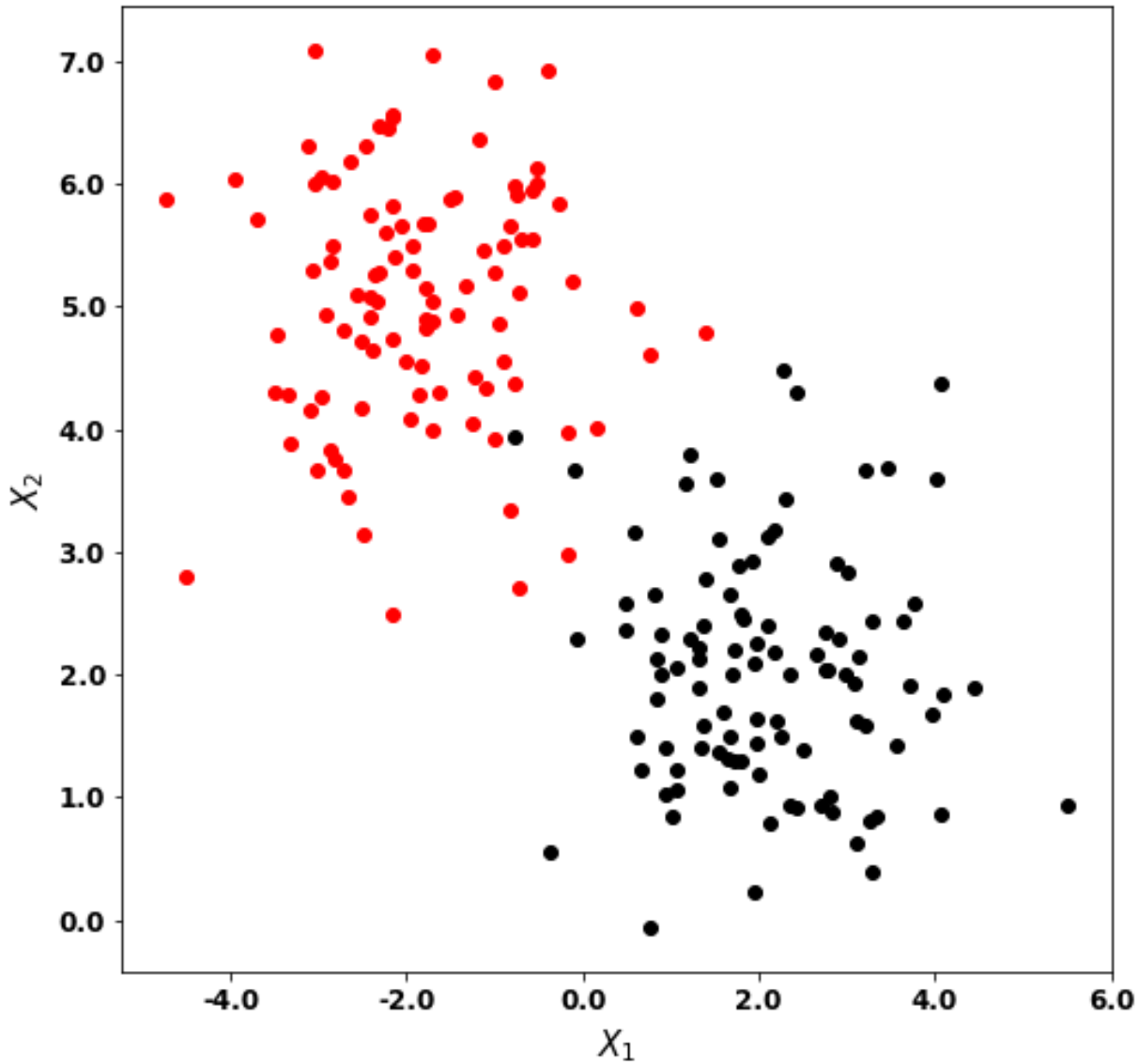
$$= \sum_{\substack{K=1 \\ K \neq K'}}^{K} \hat{p}_{mK}\, \hat{p}_{mK'}\, L_{KK'}$$

Instability (Unstable Classifier)

- Decision tree
  - Sensitive to training data
  - Small sample size
    - High variance
  - Decision tree tends to overfit

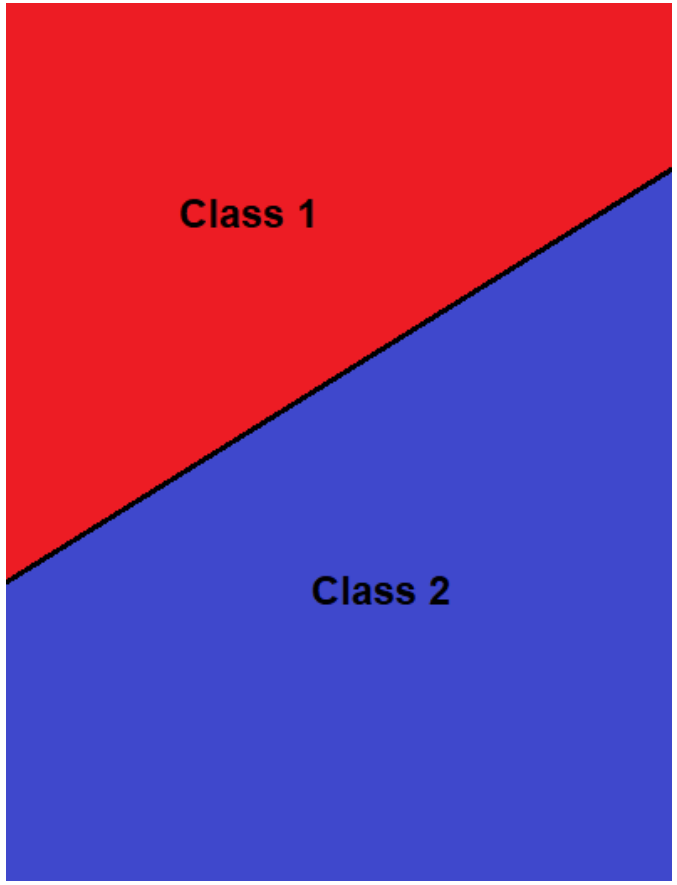
Comparing decision trees with logistic regression

Example dataset1

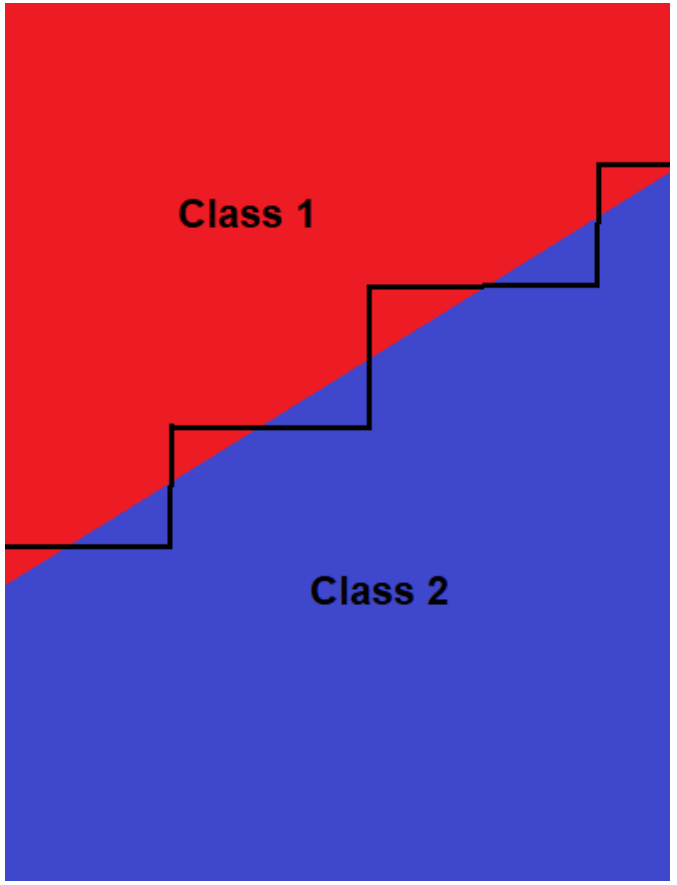


Decision trees vs logistic regression: visualising the classification in dataset1

Logistic regression

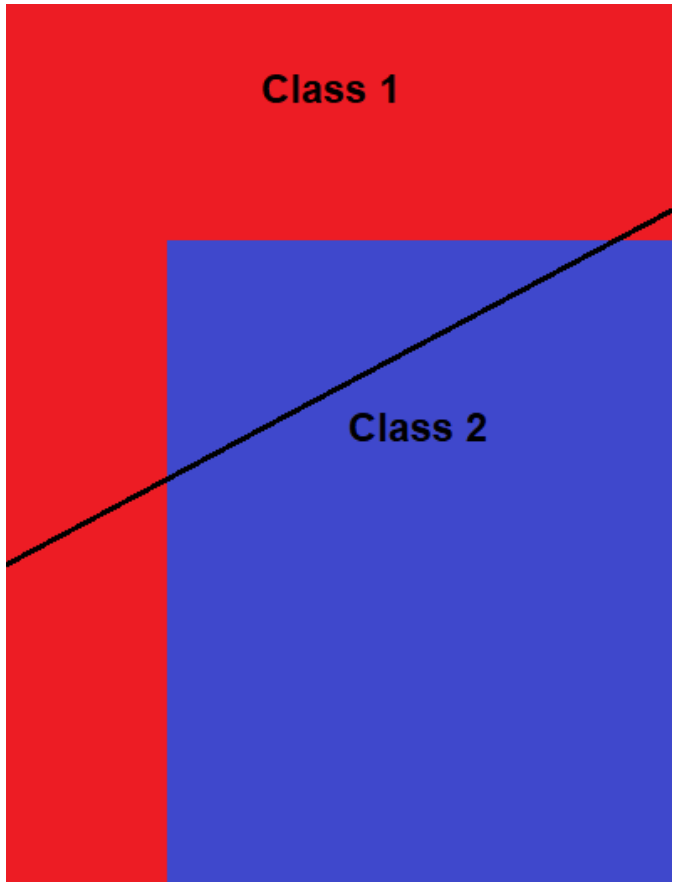


Decision trees

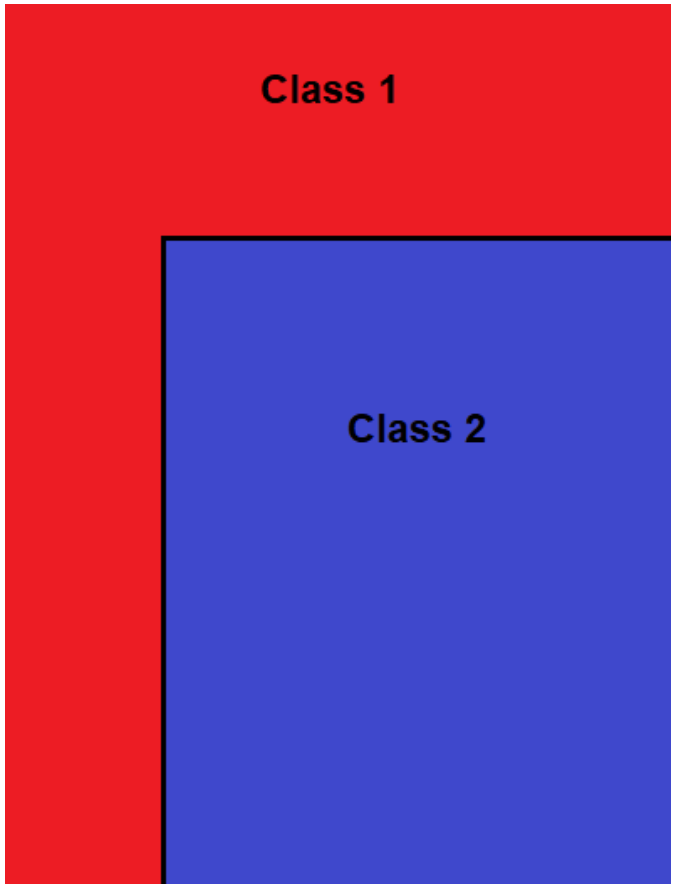


Decision trees vs logistic regression: visualising the classification in dataset2

Logistic regression



Decision trees



Example dataset1

Logistic regression

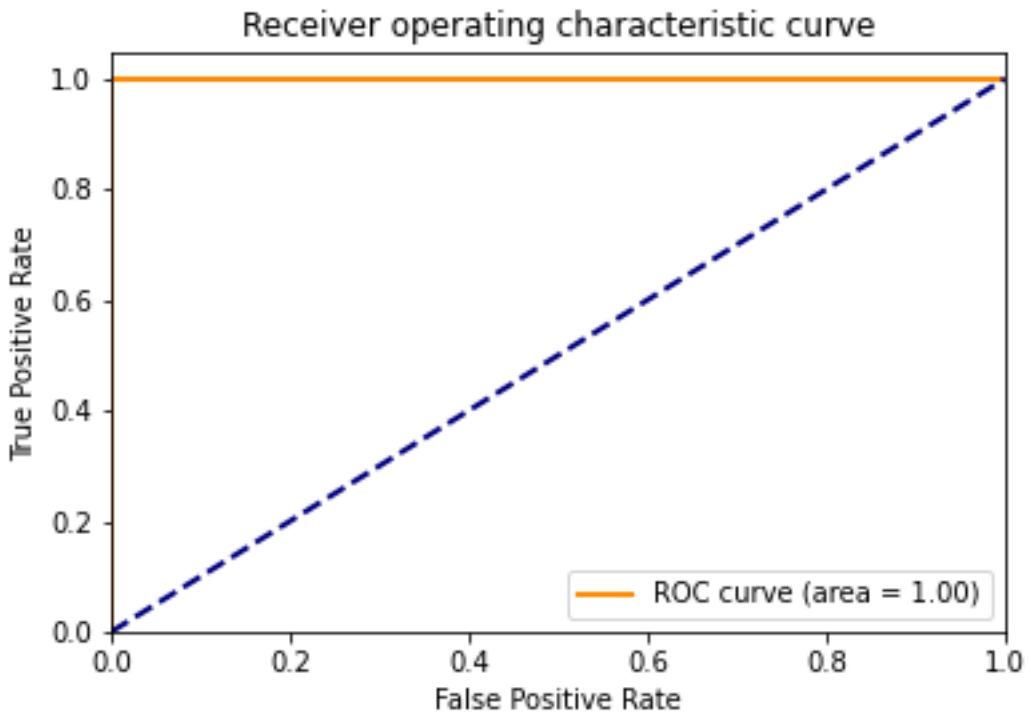
Accuracy	1
Precision	1
Recall	1
F1 score	1

Decision tree

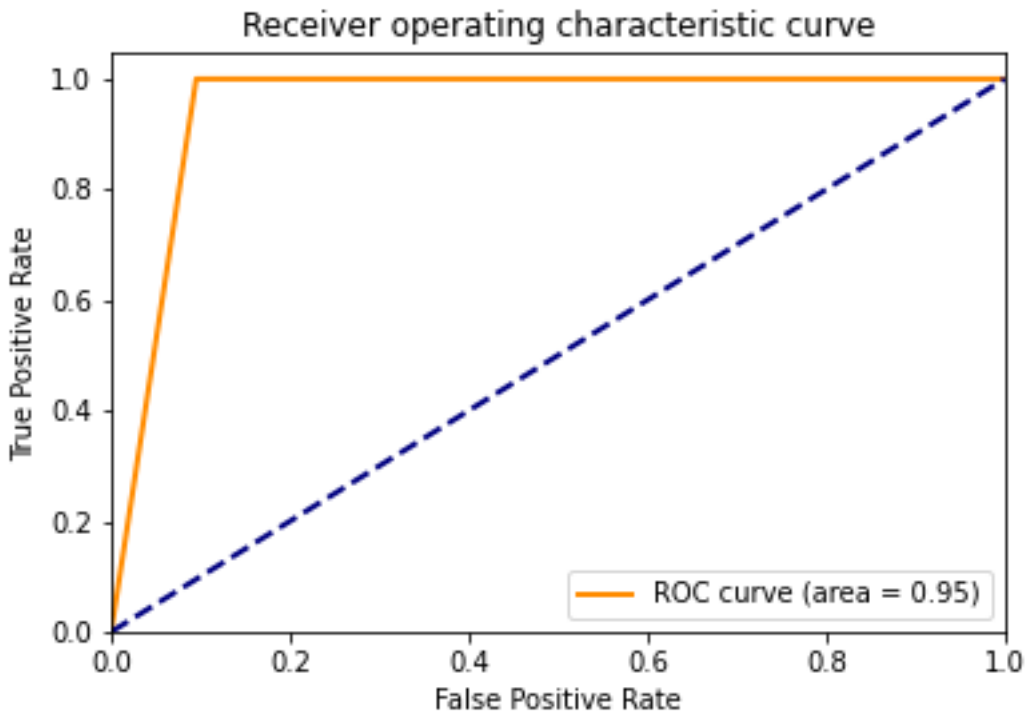
Accuracy	0.95
Precision	0.9
Recall	1
F1 score	0.95

Example dataset1

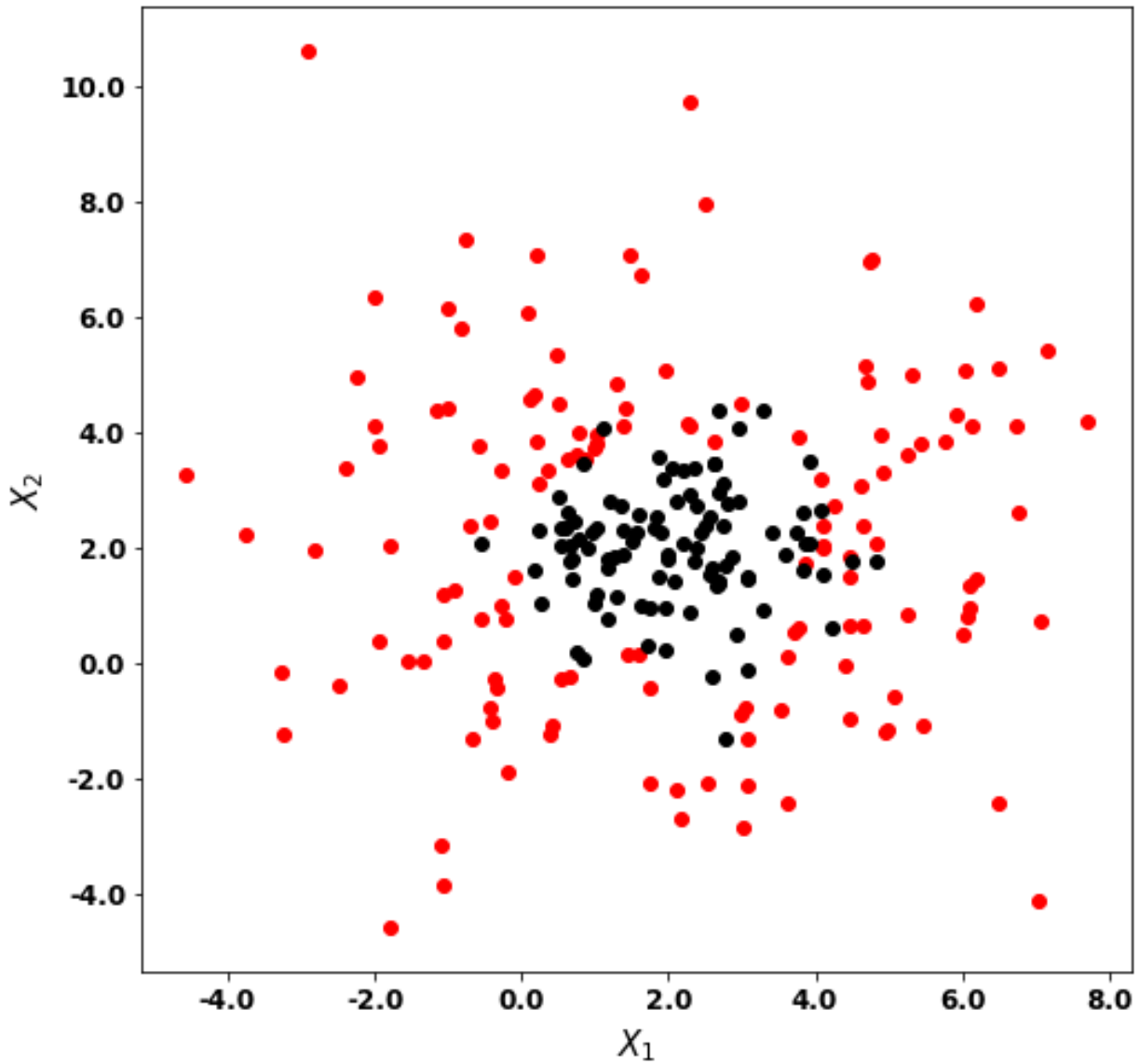
Logistic regression



Decision tree



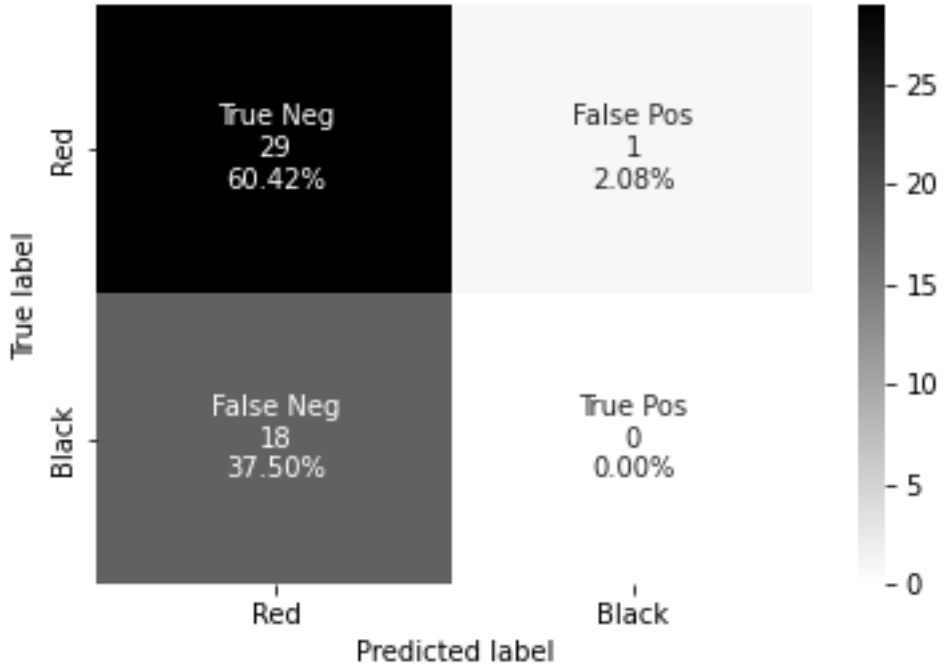
Example dataset2



Example dataset2

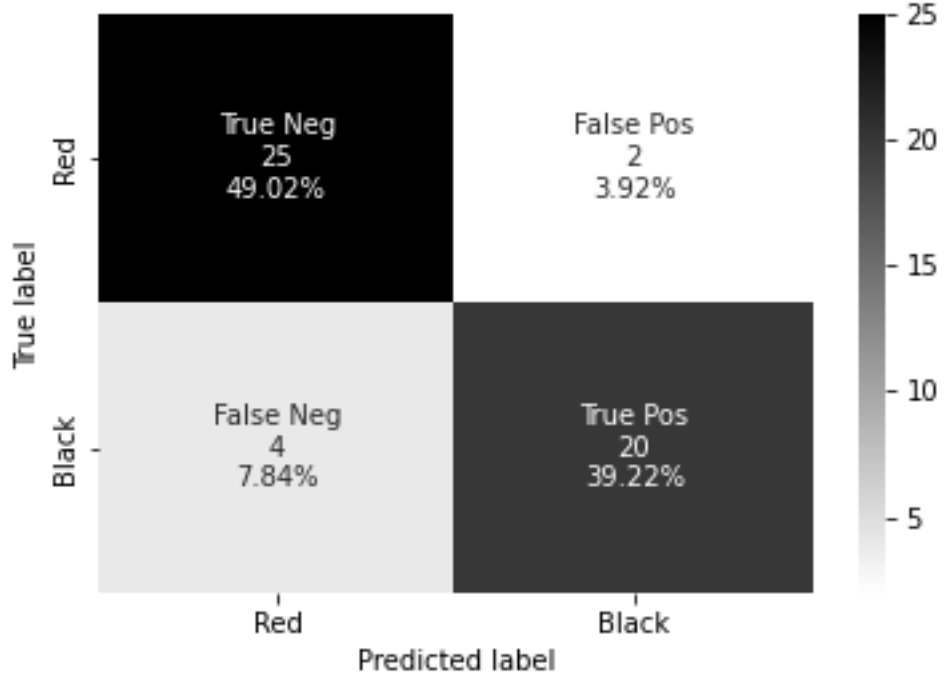
Logistic regression

Accuracy	0.6
Precision	0
Recall	0
F1 score	NaN



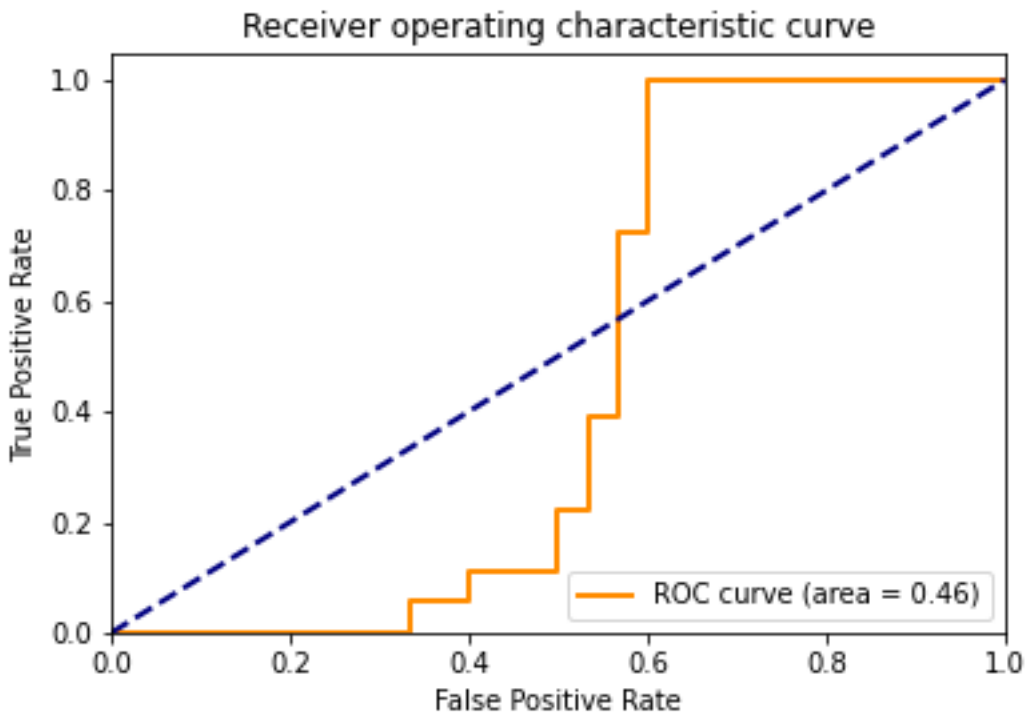
Decision tree

Accuracy	0.88
Precision	0.91
Recall	0.83
F1 score	0.87

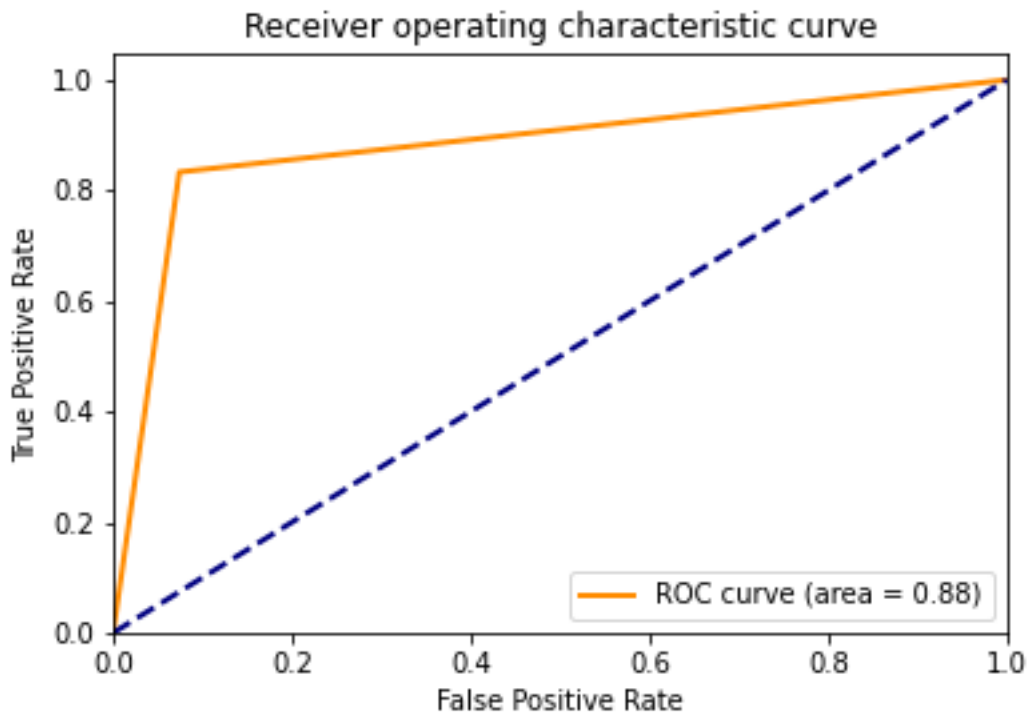


Example dataset2

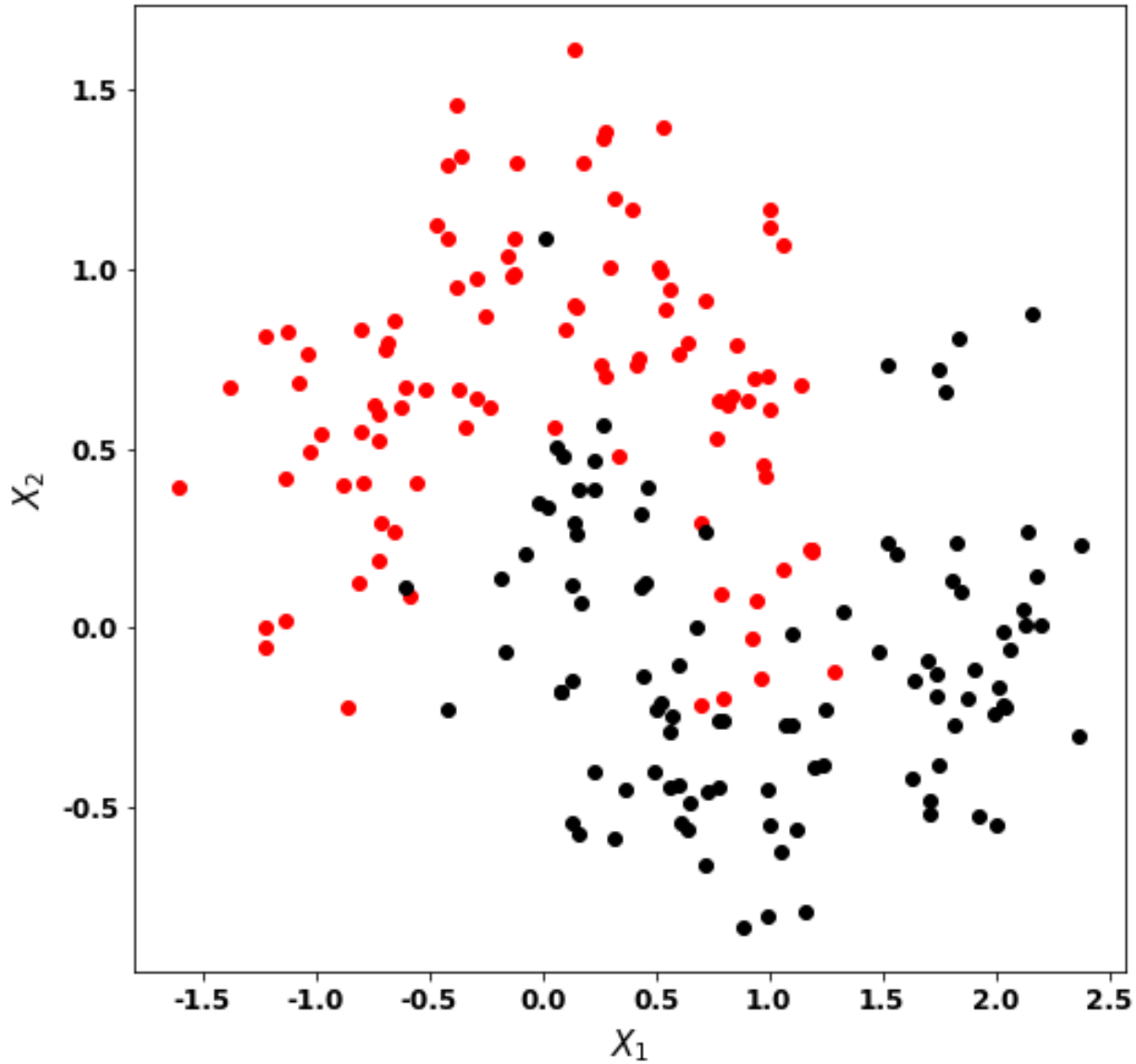
Logistic regression



Decision tree



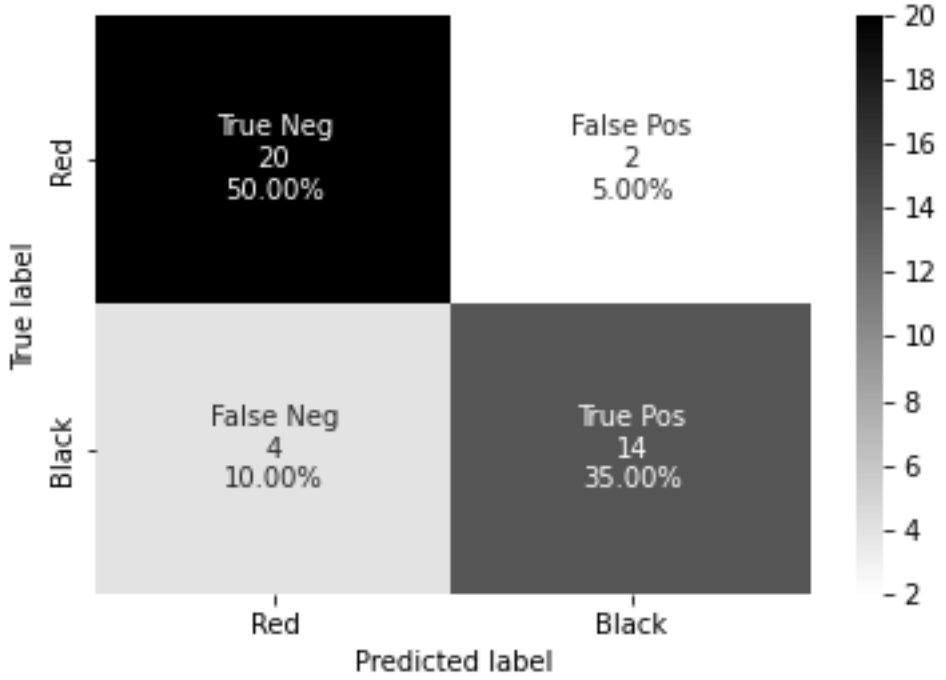
Example dataset3



Example dataset3

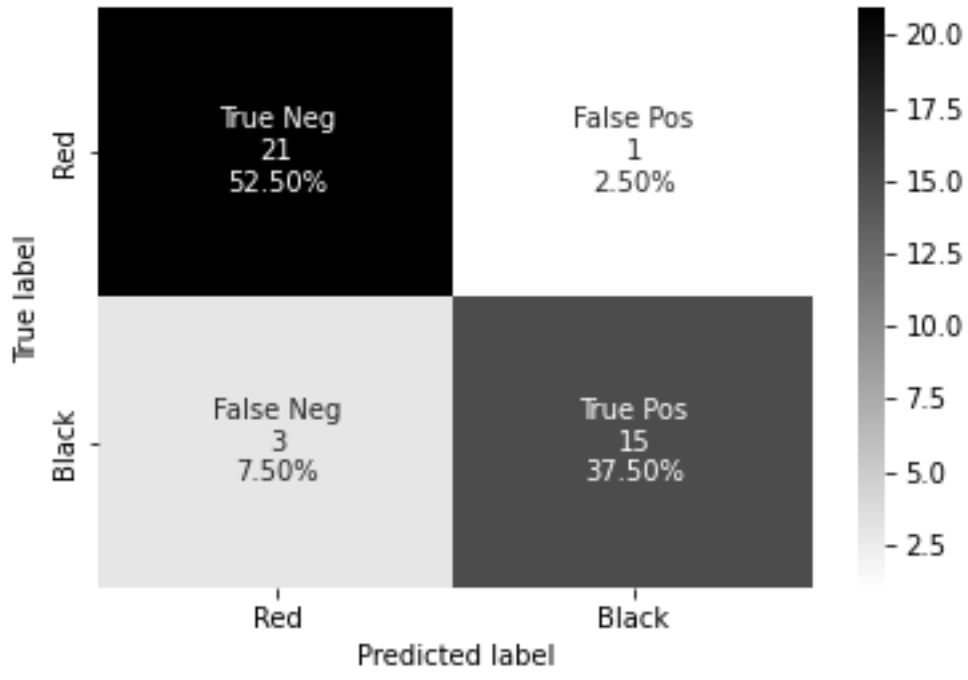
Logistic regression

Accuracy	0.85
Precision	0.88
Recall	0.78
F1 score	0.82



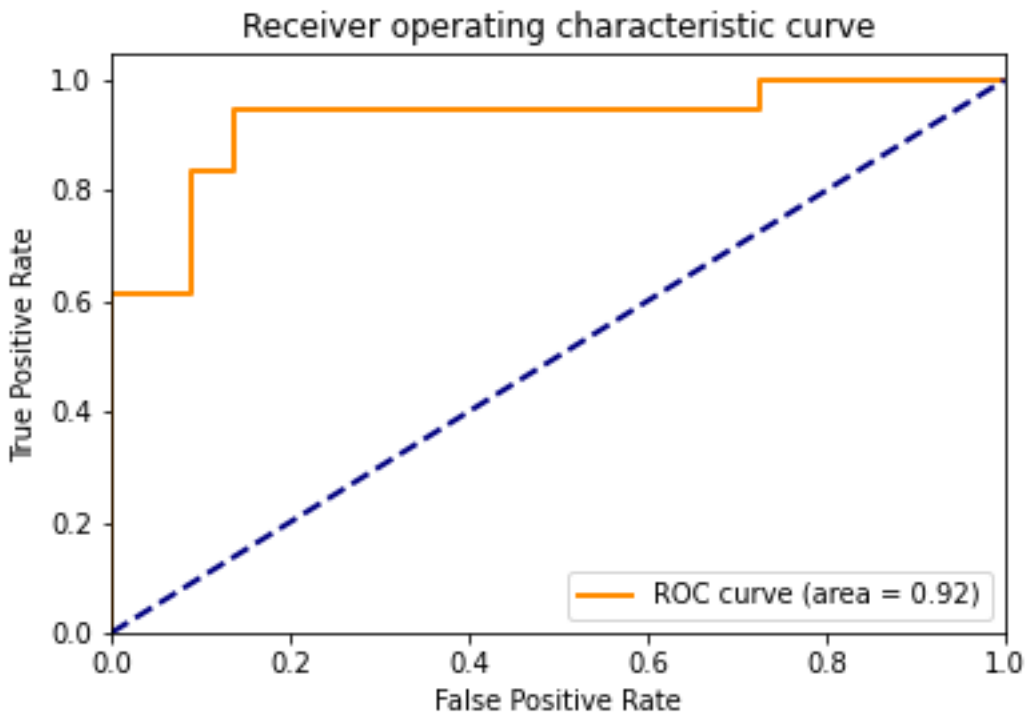
Decision tree

Accuracy	0.9
Precision	0.94
Recall	0.83
F1 score	0.88

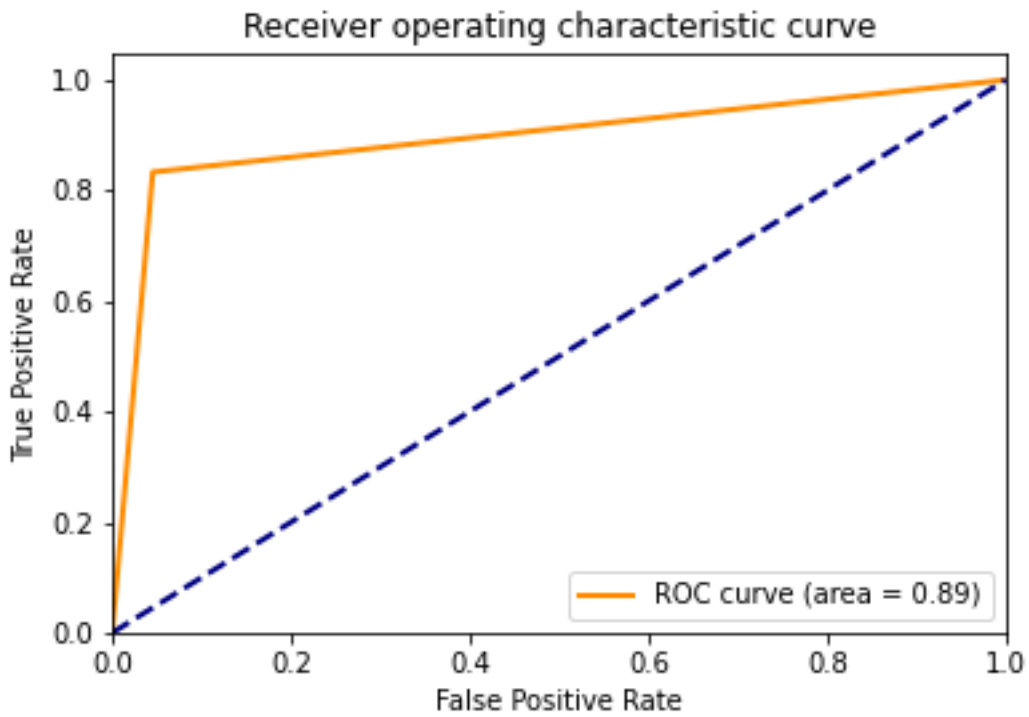


Example dataset3

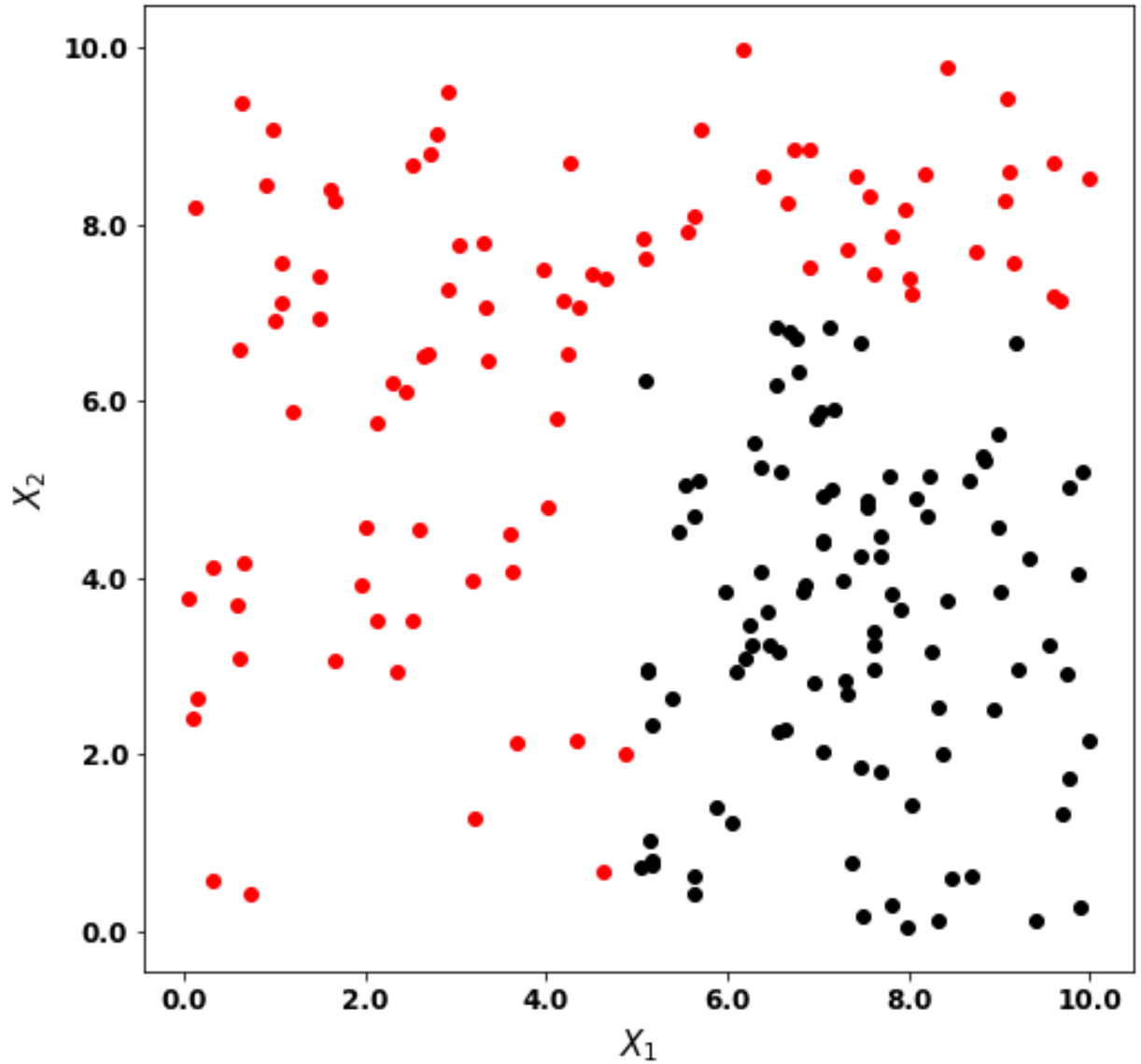
Logistic regression



Decision tree



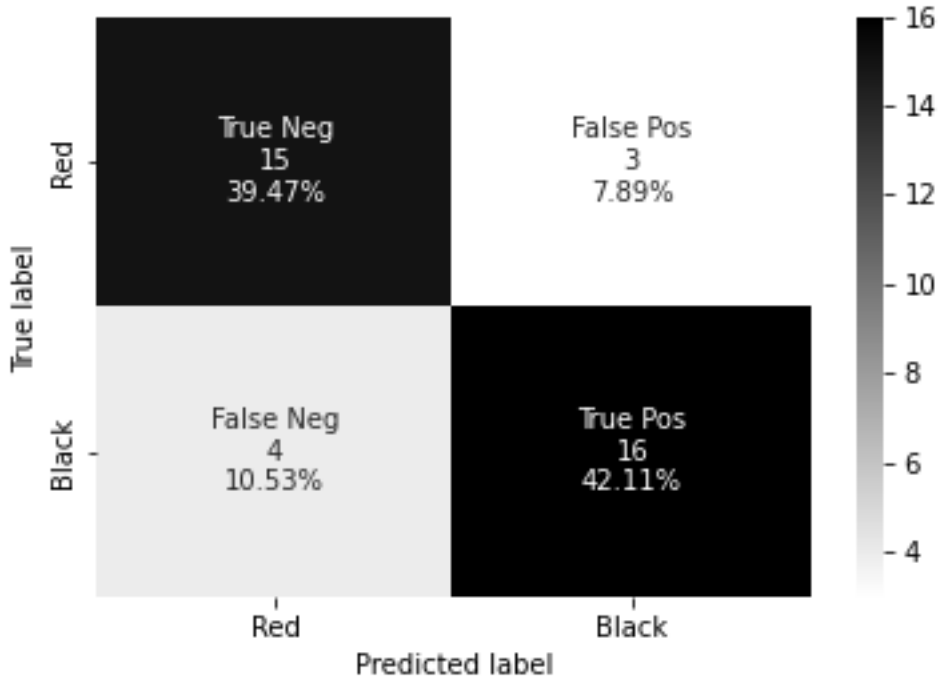
Example dataset4



Example dataset4

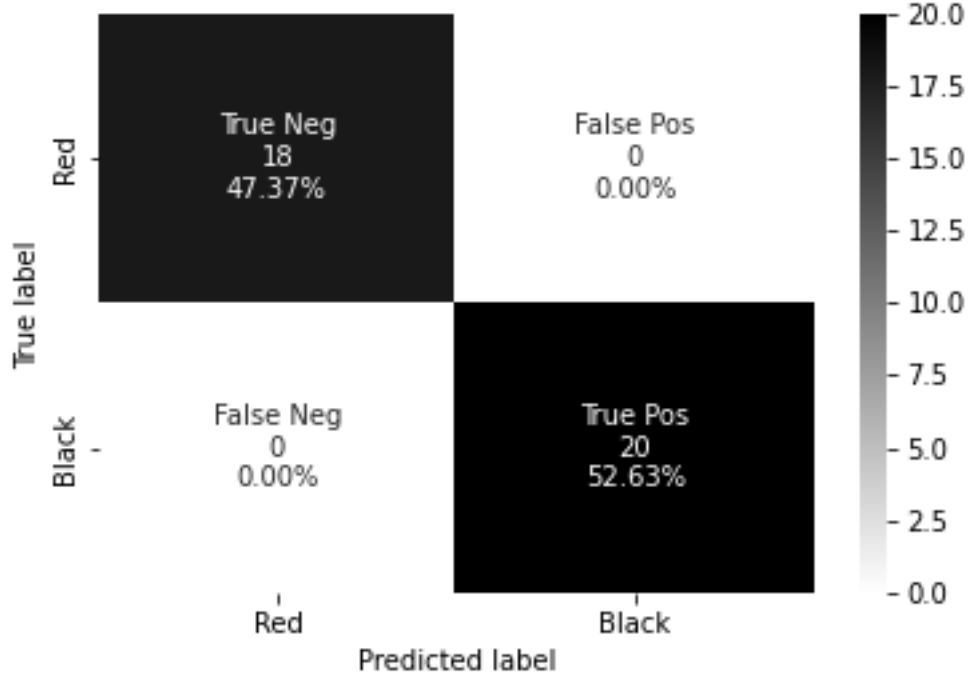
Logistic regression

Accuracy	0.82
Precision	0.84
Recall	0.8
F1 score	0.82



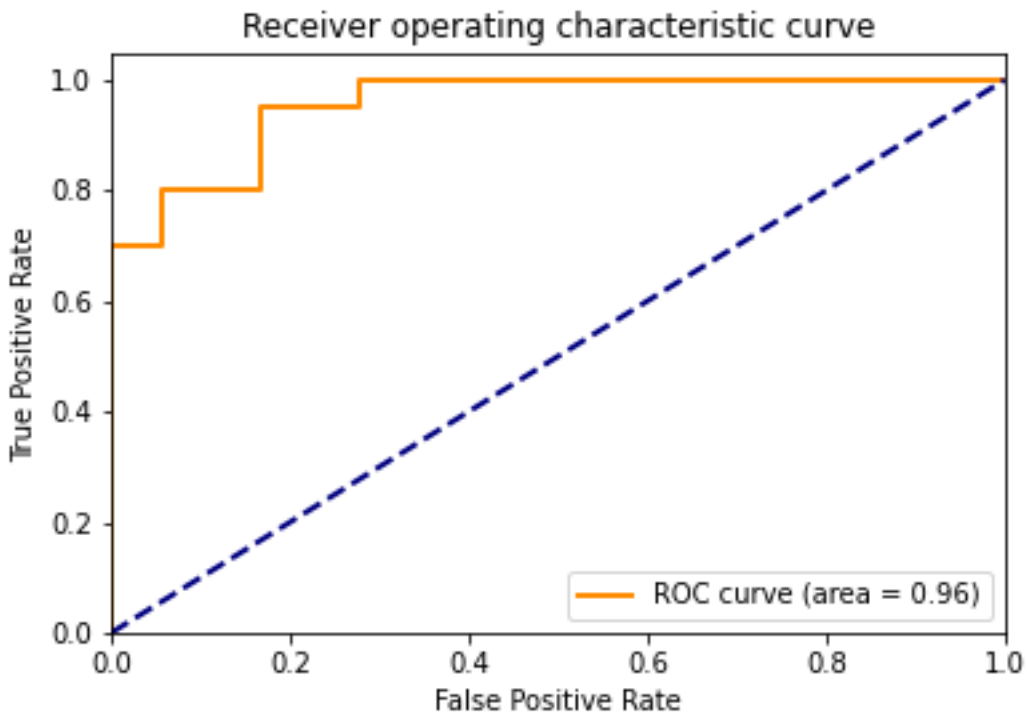
Decision tree

Accuracy	1
Precision	1
Recall	1
F1 score	1

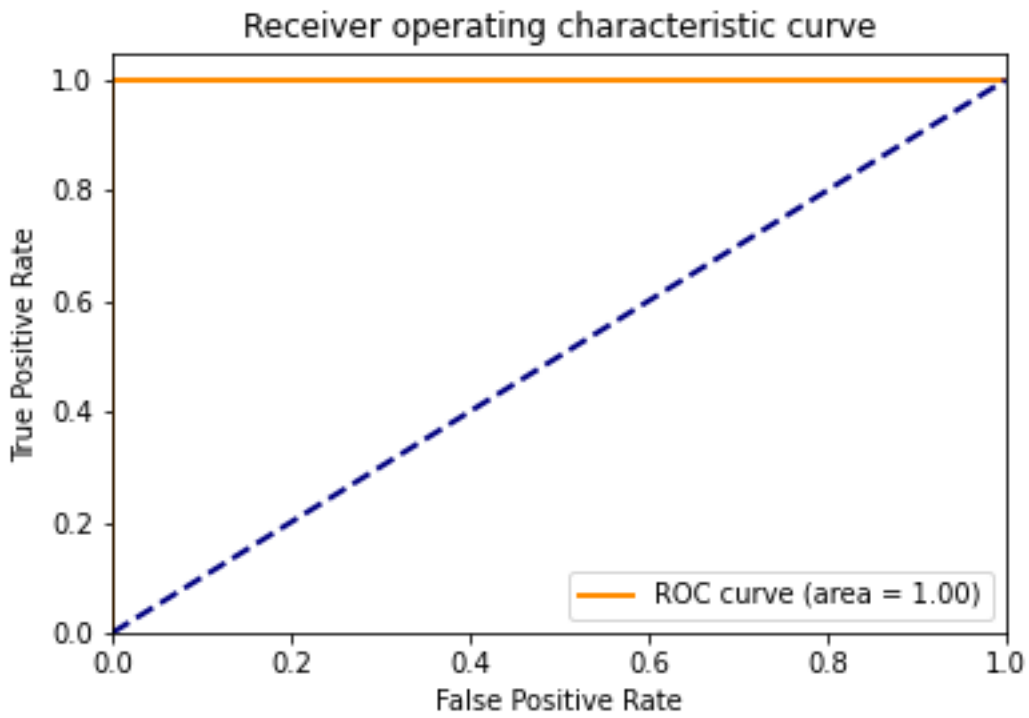


Example dataset4

Logistic regression



Decision tree

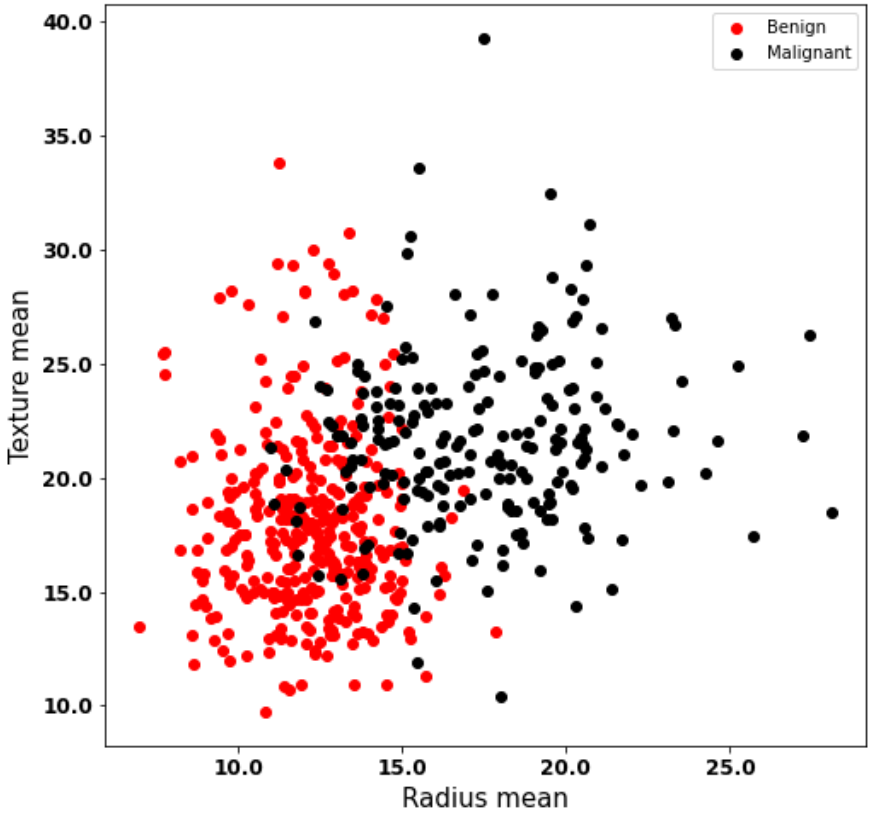


Cancer geometry data

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

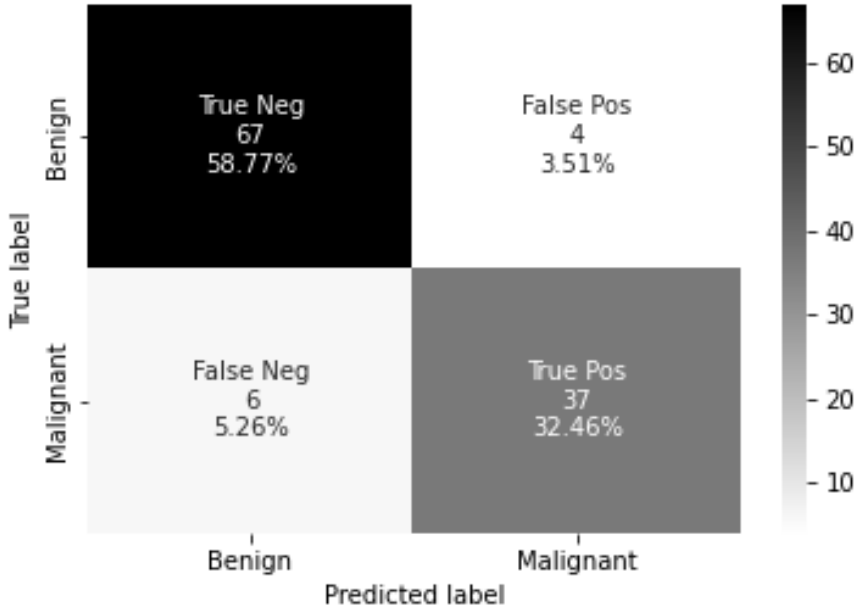
Features used for classification: radius mean and texture mean



Cancer geometry data

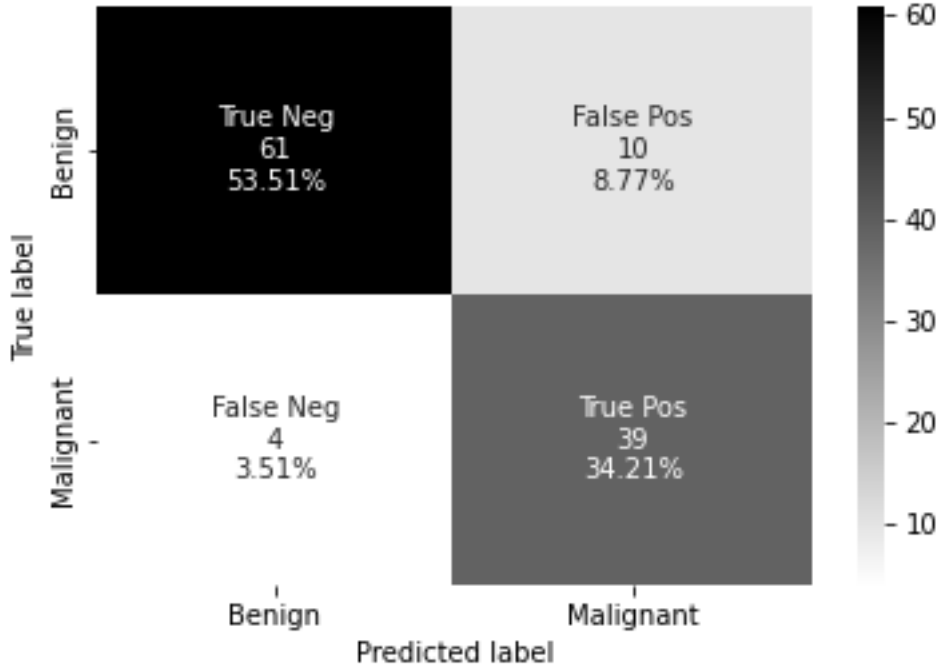
Logistic regression

Accuracy	0.91
Precision	0.90
Recall	0.86
F1 score	0.88



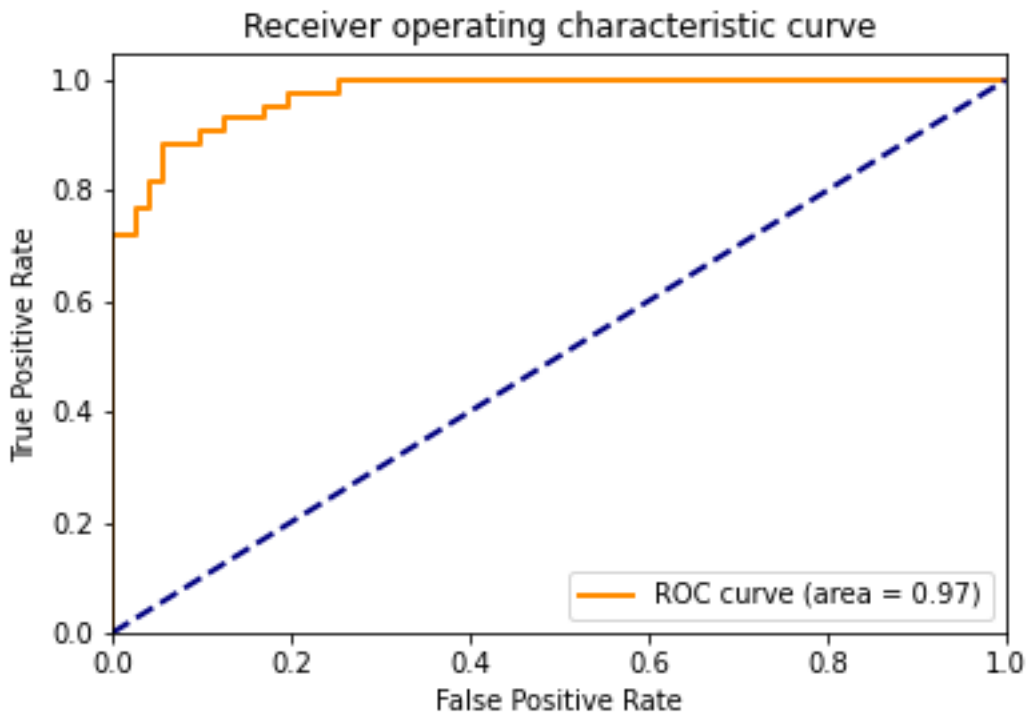
Decision tree

Accuracy	0.88
Precision	0.8
Recall	0.91
F1 score	0.85

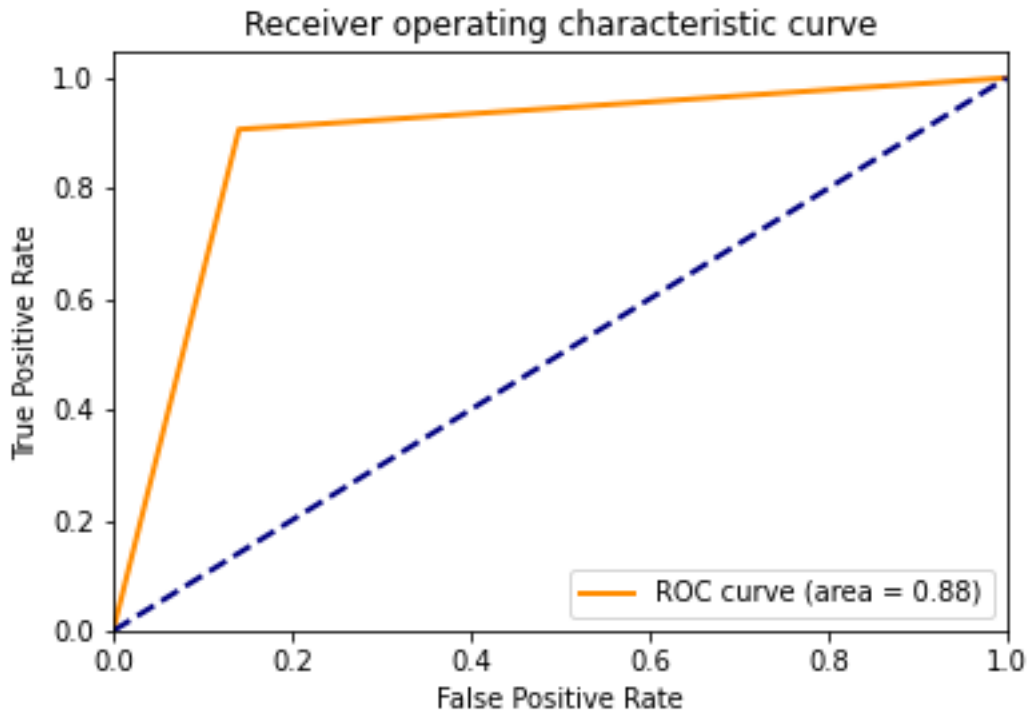


Cancer geometry data

Logistic regression



Decision tree



Bagging

Decision trees: High variance model

- Randomly generated dataset (say D1) is split to two datasets (say D1_1 and D1_2) with equal number of samples in each of them.
- Decision tree classifiers are trained on each of them.

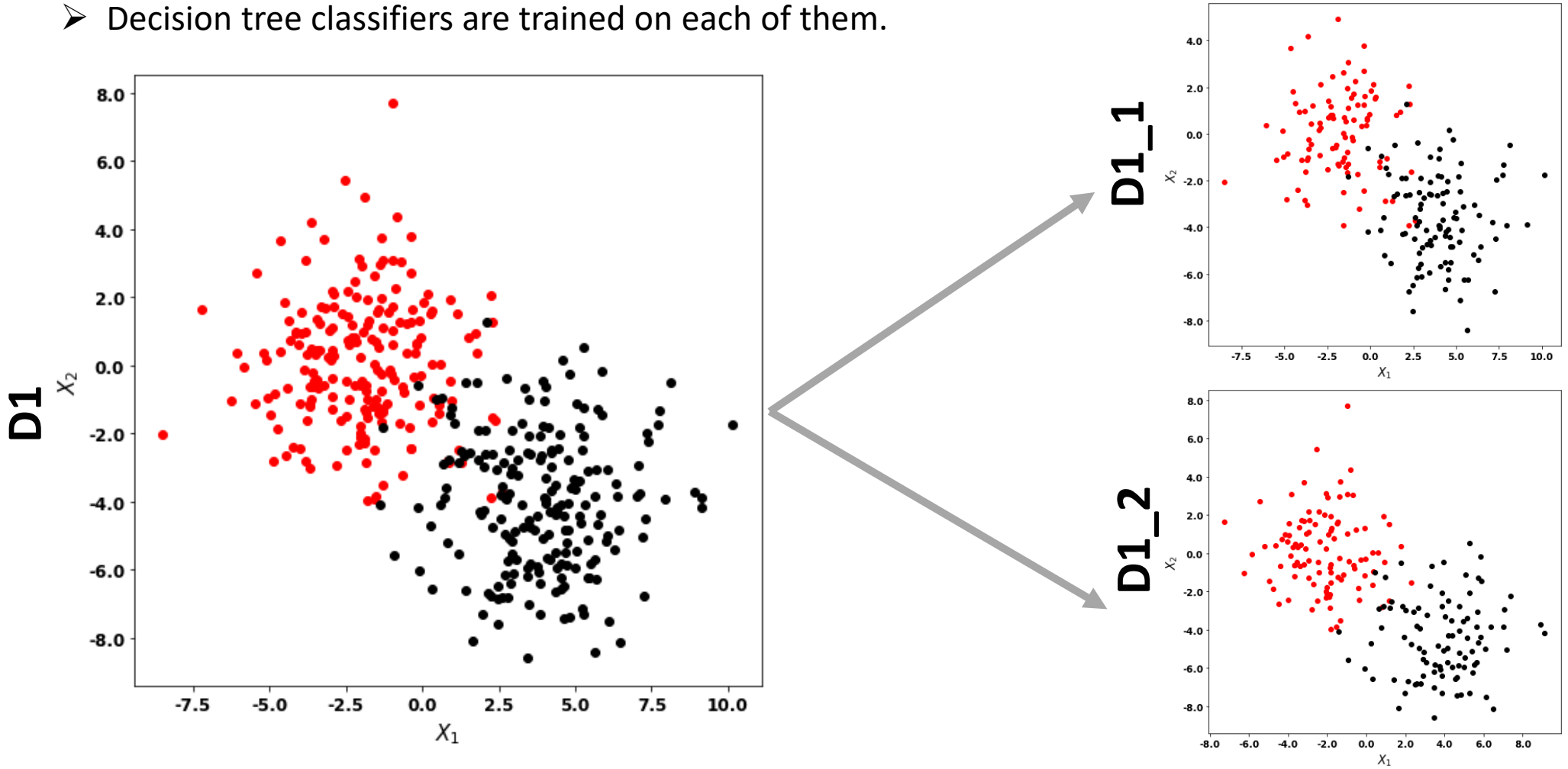


Figure1: A large dataset with 400 samples is divided to two datasets with same number of samples

Decision trees: High variance model

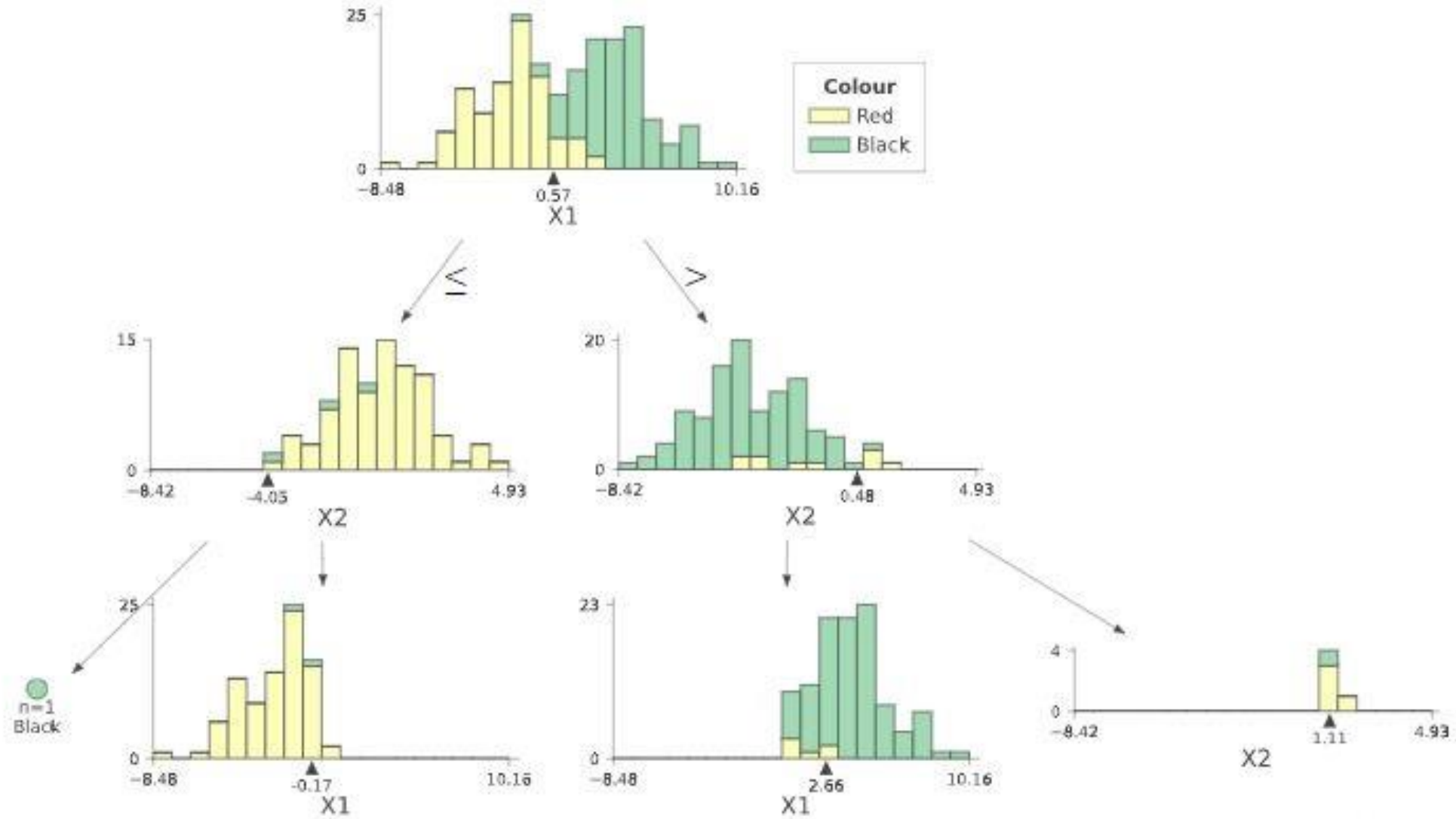


Figure2: Splits made by decision tree trained on D1_1 visualized till depth 3.

Decision trees: High variance model

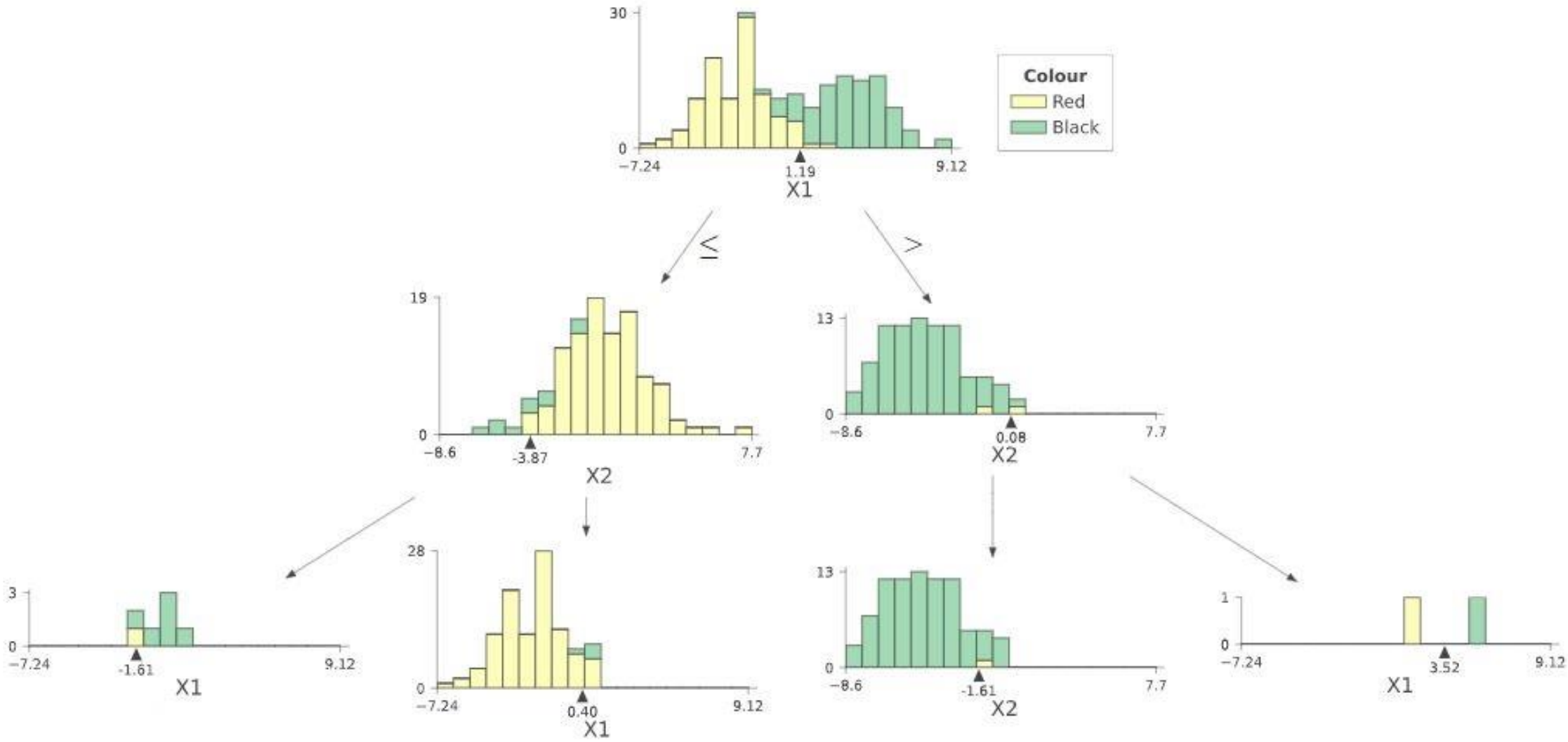


Figure3: Splits made by decision tree trained on D1_2 visualized till depth 3.

How to overcome overfitting/high variance?

- One way to combat overfitting is by generating lots and lots of dataset and fitting a decision tree on each of them.
- The predicted class would be the one that has maximum number of votes from the decision trees.

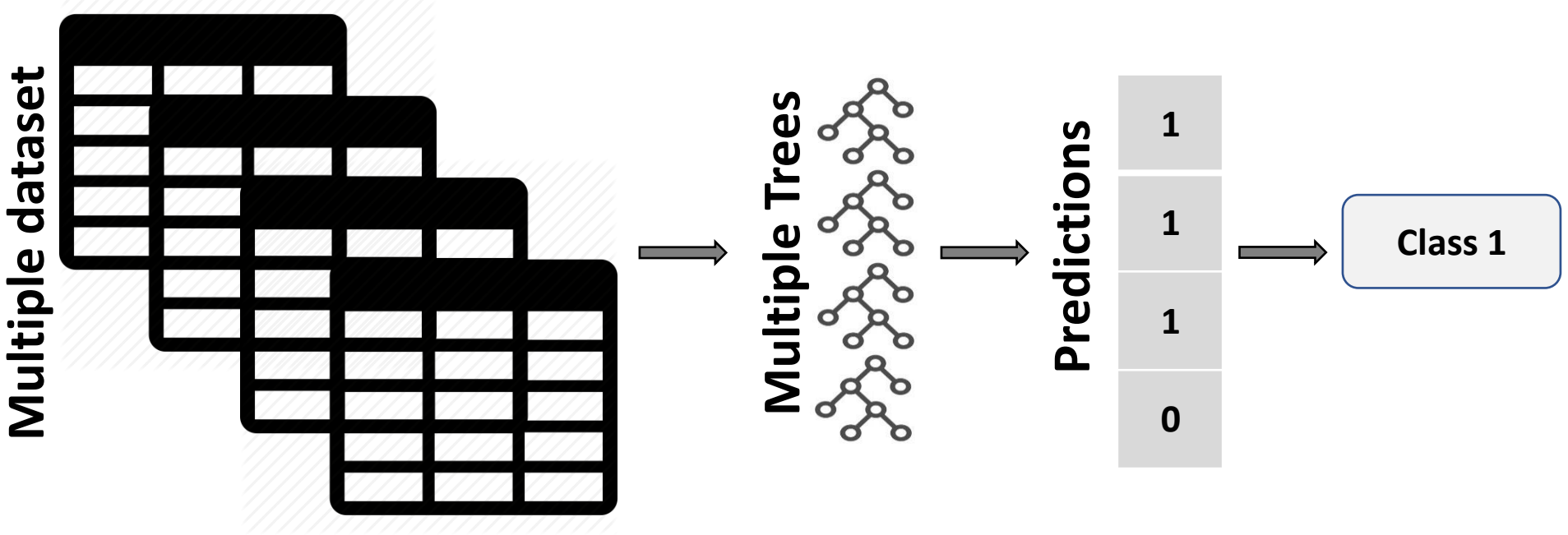


Figure4: We can overcome high variance by fitting multiple trees on multiple datasets

But where can we get these large amounts of data?

Bootstrapping

Bootstrap is a method to generate artificial datasets by sampling without replacement from the original dataset

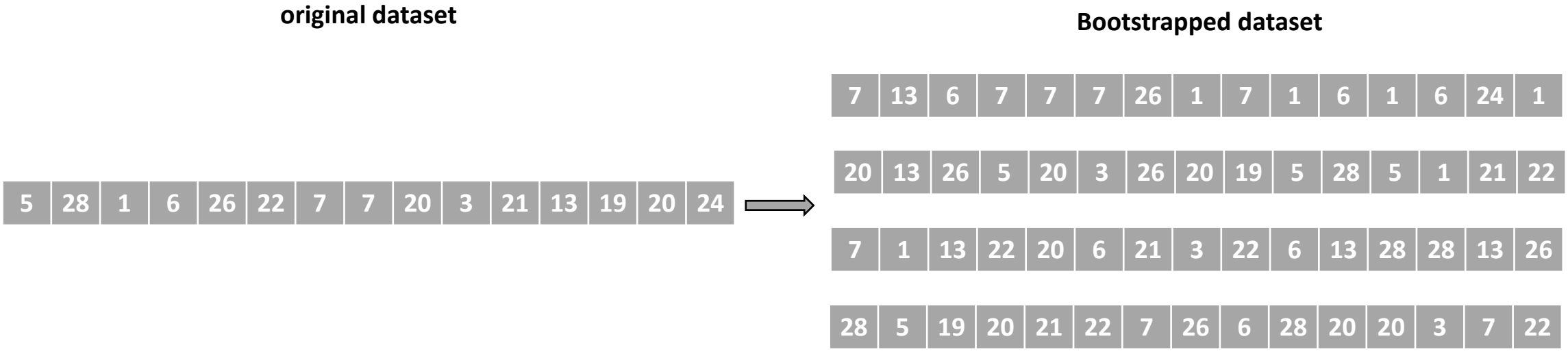
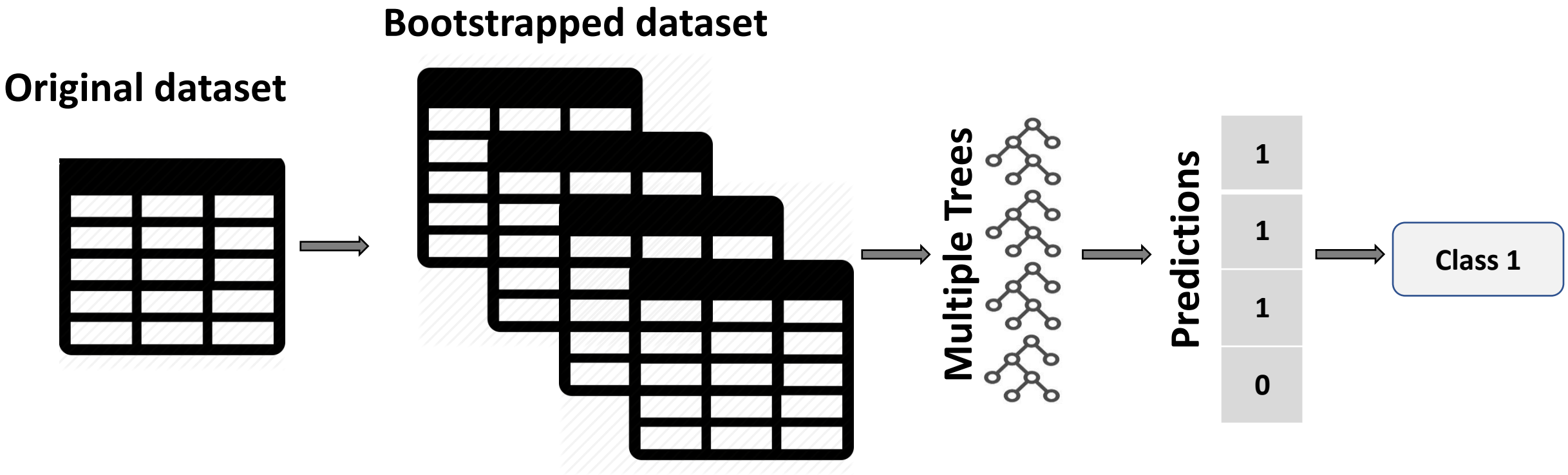


Figure5: Bootstrapping to get replicates of the original dataset

Bagging classifier

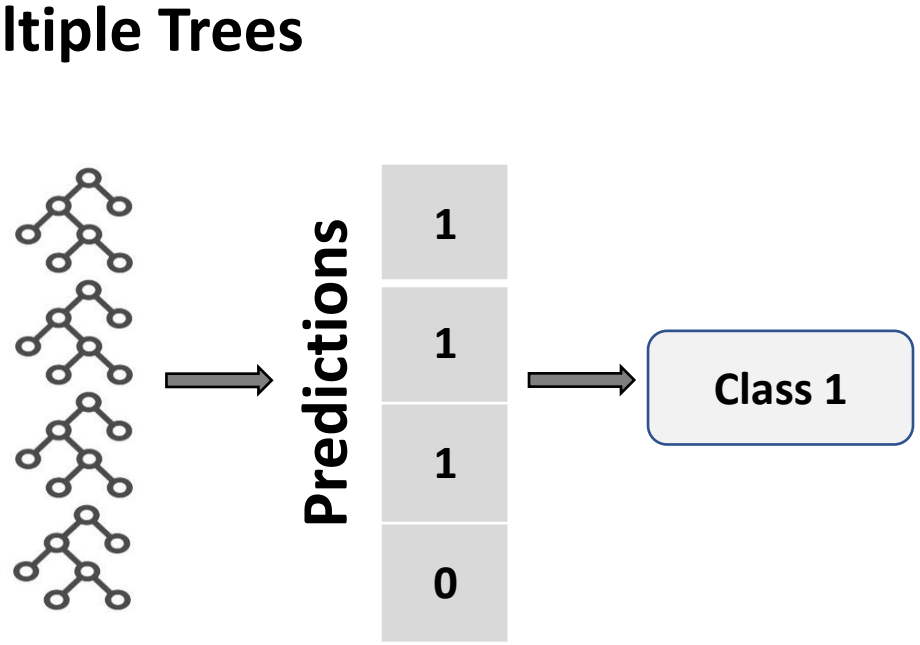
Bootstrap aggregation (or) Bagging



Prediction: Averaging

Bootstrap aggregation (or) Bagging

- Individual trees have high variance and low bias
- Bagged trees have low variance.



Lost the interpretability advantage of decision trees.

Out of bag error (OOB error)

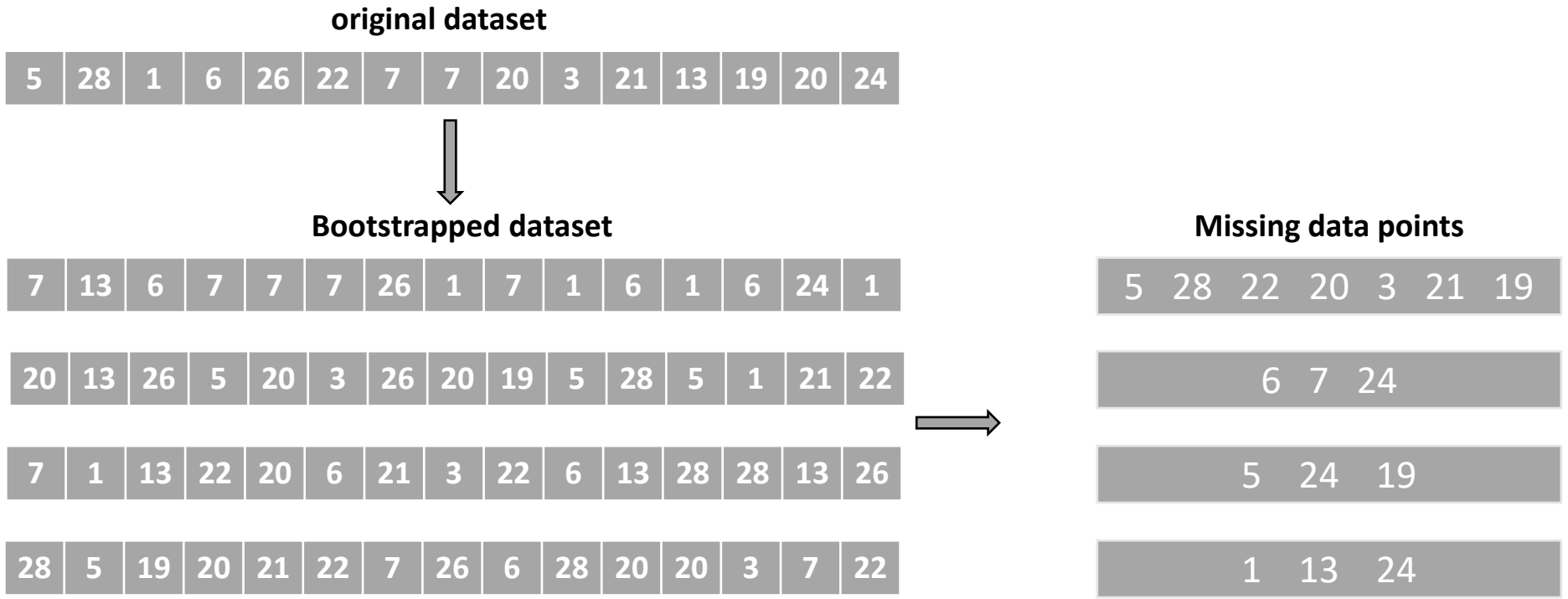


Figure7: We will miss few data points for sure in each of the bootstrapped data

- Each of the trees in bagging classifier are not trained with all data points in the original dataset.
- The missed data points can be used as validation test data (out of bag error) tune hyper parameters or model selection.
- With size of bootstrap increasing OOB error becomes virtually equal to leave-one-out cross-validation error.

Out of bag error (OOB error)

- Each of the trees in the bagging classifier is not trained with all data points in the original dataset
- The missed data points can be used as validation test data (out of bag error) tune hyperparameters or model selection
- With the size of bootstrap increasing OOB error becomes virtually equal to leave-one-out cross-validation error.

Bagging classifier: correlated trees?

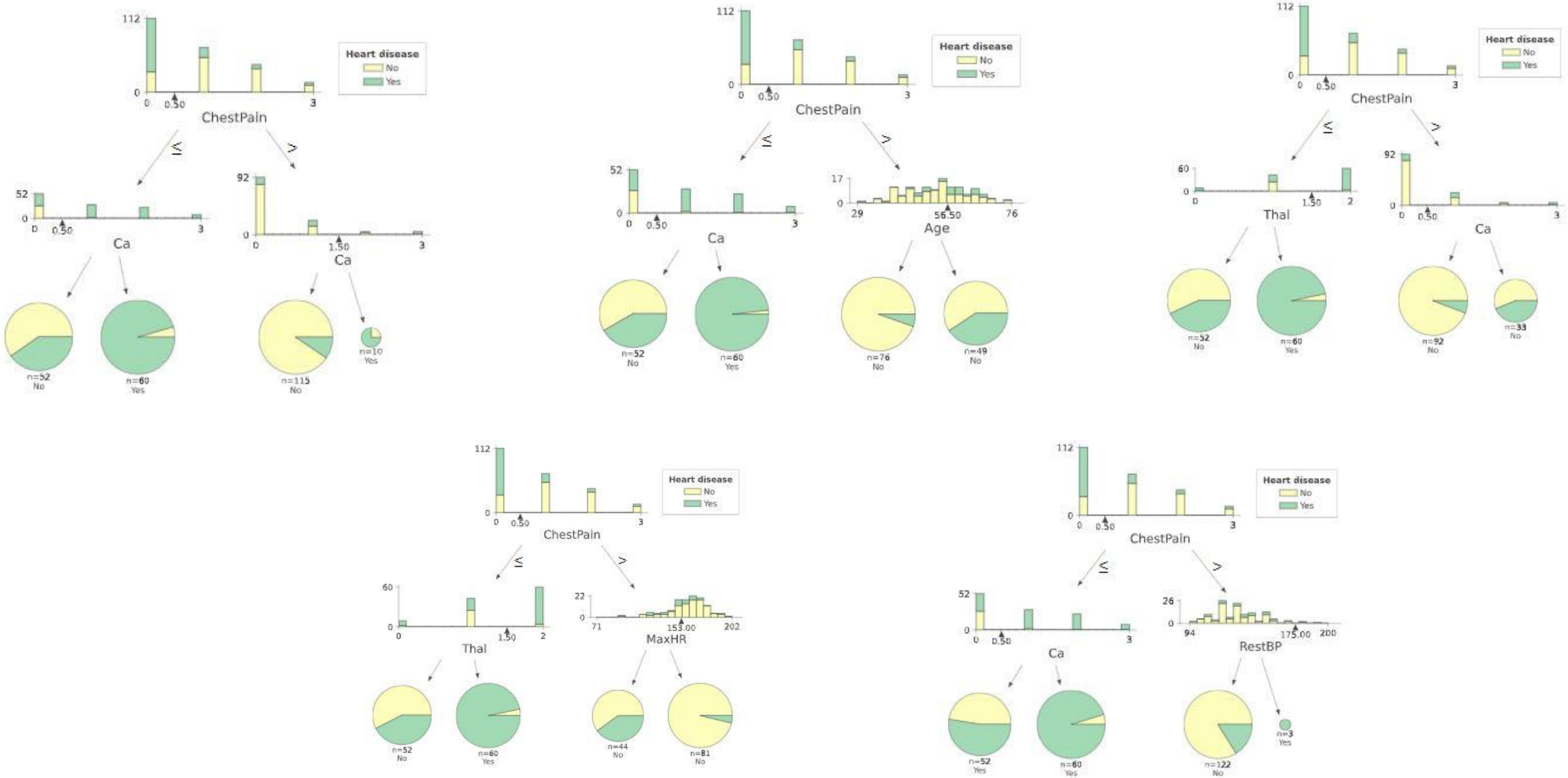


Figure8: Five randomly chosen trees with depth=2 in a bagging classifier for heart disease dataset

How to overcome the correlation between the trees?

- Presence of strong feature (chest pain in heart data) makes the tree to choose it as a root node.
- This brings correlation among the trees and hence most trees look similar in bagging

Why not use randomly chosen features for training each of the tree

Boosting

Boosting

- Boosting differs from bagging and random forests by growing trees sequentially.
- Each tree is trained on the modified dataset and grown on information from previous trees.
- Works by learning from ‘mistakes’ made in past.

* Each stage add one more classifier such that it reduces error
* ADABOOST: Adaptive boosting
* Gradient boosting

Boosting ADABOOST

K-1 stage classifier

$$C_{K-1}(x_i) = \alpha_1 M_1(x_i) + \alpha_2 M_2(x_i) + \cdots \alpha_{K-1} M_{K-1}(x_i)$$

K^{th} stage classifier

$$C_K(x_i) = C_{K-1} M_{K-1}(x_i) + \alpha_K M_K(x_i)$$

M_K: K^{th} stage model

$M_K(x_i)$: Train model such that residual error decreases.

Boosting

Loss function: $e^{-y_i f(x_i)}$

$$\text{Loss}(K) = \sum_{i=1}^{N} e^{-y_i C_K(x_i)}$$

$$= \sum_{i=1}^{N} e^{-y_i [C_{K-1}(x_i) + \alpha_K M_K(x_i)]}$$

$$= \sum_{i=1}^{N} e^{-y_i C_{K-1}(x_i)} \; e^{-y_i \alpha_K M_K(x_i)}$$

$e^{-y_i C_{K-1}(x_i)}$ → W_i^K : Loss incurred on the i^{th} point after $(K-1)$ stage

$$= \sum_{i=1}^{N} W_i^K \; e^{-y_i \alpha_K M_K(x_i)}$$

Boosting

→ $y_i = M_K(x_i) = 1$

→ $y_i = 1$, $M_K(x_i) = -1$

Loss

$$= \sum_{y_i = M_K(x_i)} w_i^K e^{-\alpha_K} + \sum_{y_i \neq M_K(x_i)} w_i^K e^{+\alpha_K}$$

$$= e^{-\alpha_K} W_c + e^{\alpha_K} W_e$$

$W_c + W_e = W = \text{const.}$

* Assign "weight to each data point"

* Boosting: Generate a new sample using the prediction error from the previous stage.

* M_K : → Estimates

* How to choose α_K

$$\frac{\partial L}{\partial \alpha_K} = 0 \Rightarrow \alpha_K^* = \frac{1}{2} \ln\left(\frac{1 - \frac{W_e^K}{W}}{\frac{W_e^K}{W}}\right)$$

M_K^{th} → classifier

* Update weights

$$w_i^{K+1} = w_i^K e^{-y_i \alpha_K M_K(x_i)}$$

Boosting

Boosting algorithm (from Introduction to statistical learning):

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all *i* in the training set.
2. For b = 1, 2, . . .,B, repeat:
   a) Fit a tree $\hat{f}^b$ with d splits (d+1 terminal nodes) to the training data (X, r).
   b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

   c) Update the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$

3. Output of the boosted model

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Boosting

Key points:

- We will use regression trees* for boosting classifier.
- Note that in boosting increasing number of trees (B) beyond certain number leads to overfitting unlike the case of bagging and random forests.
- B has to be tuned using validation or cross-validation set.
- Shrinkage parameter, λ decides the rate of learning of boosting.
- Small value of λ requires large value for B.
- Complexity of the model is decided by number of splits, d.
- Most often d=1 (single split) works well.
- In multiclass case, multiple function are used and their output is given to softmax for prediction.

Slide #17 for regression trees

Regression trees

- Regression trees work in similar fashion of classification trees.
- Final prediction would be average of training data in leaf nodes.
- Loss/cost function is residual sum of squares.
  - $\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$ $\hat{y}_{R_j}$ ➔ Average of training data in j^{th} region
    J ➔ Total number of regions/leaf nodes

In boosting classifier:

- The outputs will be a sigmoid function of the prediction.

Random forest classifier

Random forest classifier

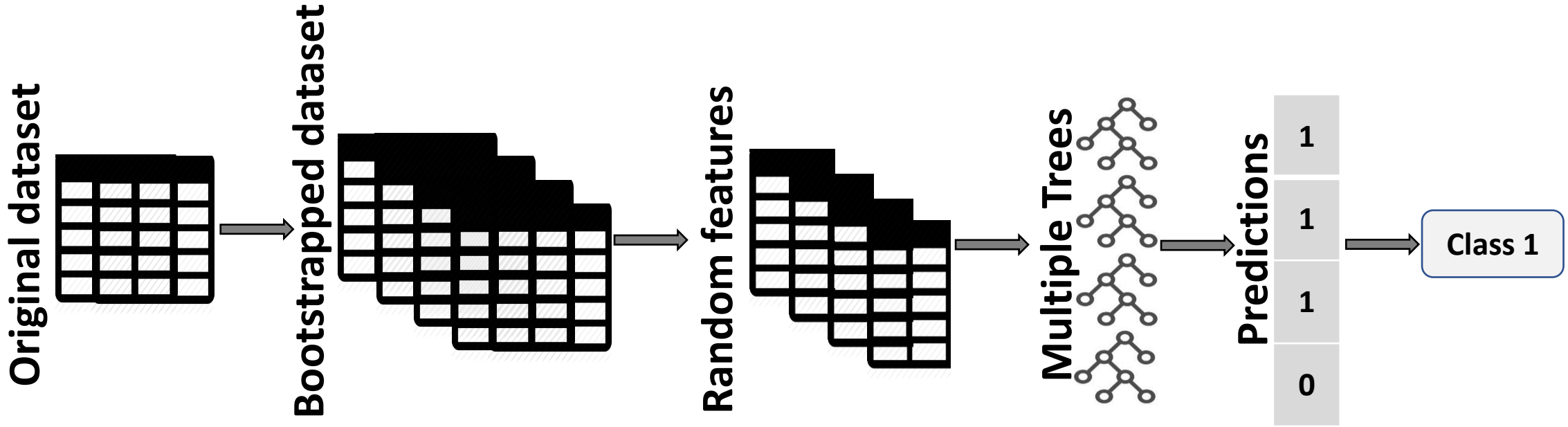


Figure9: Random forests work by choosing random features for each of the trees

- Typically, number of random features (m) chosen at each split equals to square root of total number of features (p).
- Hence in average strong predictor will not be considered in (p – m)/p splits.
- Note that bagging is a special case of random forest classifier (when m = p).

Random forest classifier trees

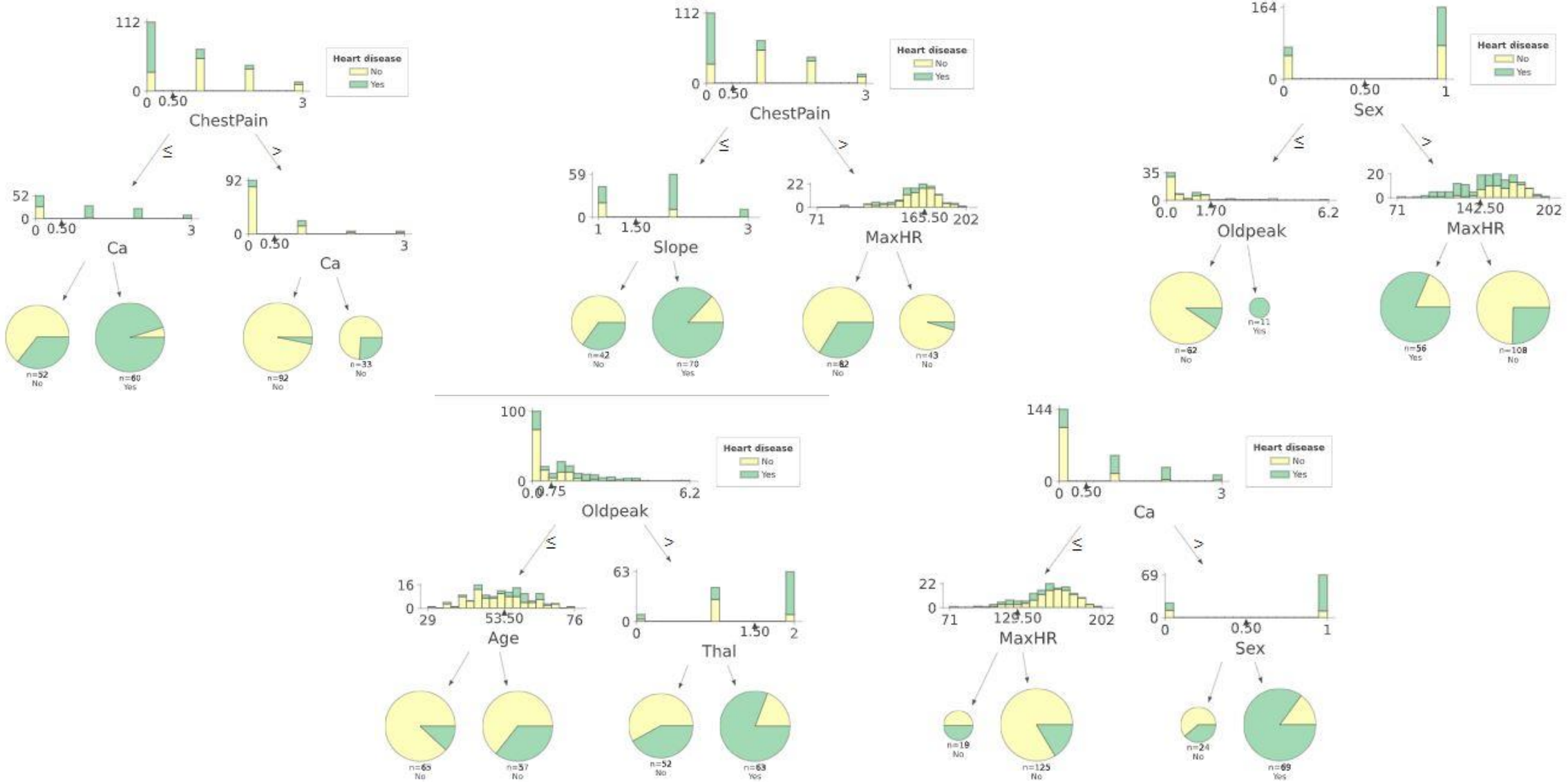
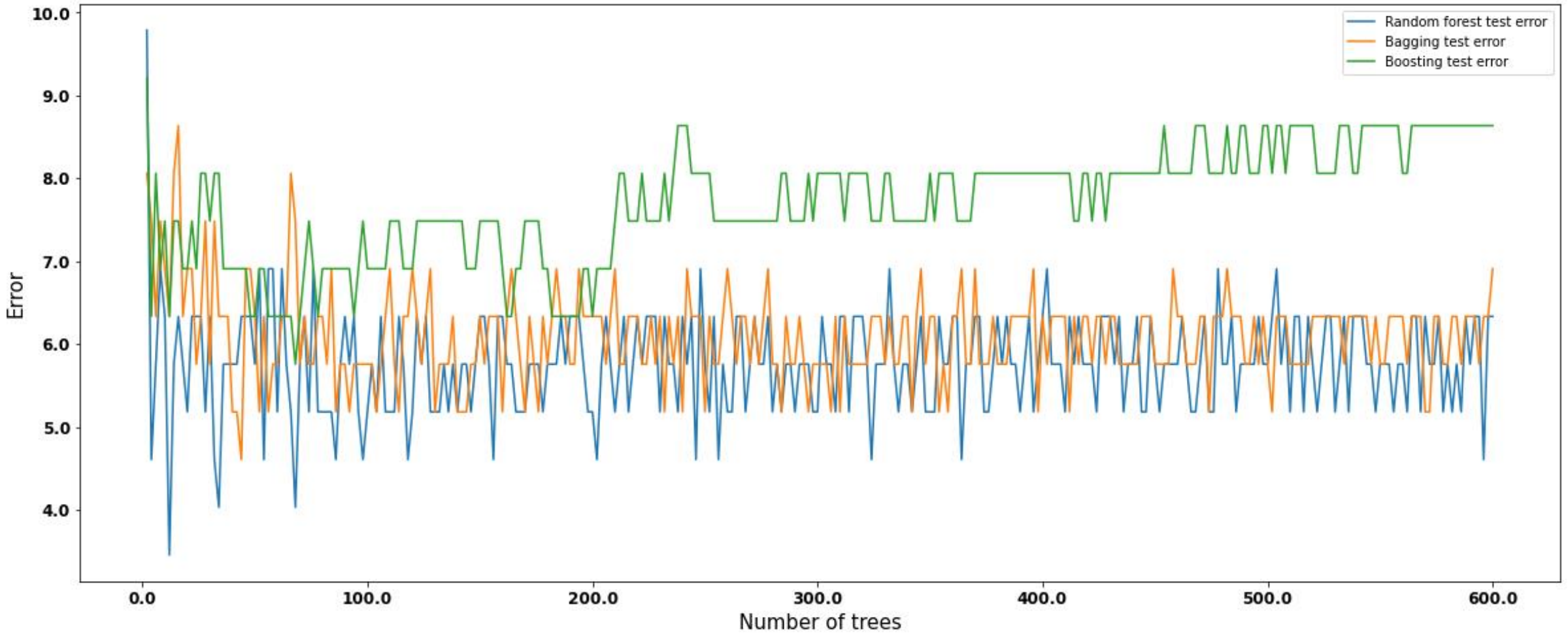


Figure10: Five randomly chosen trees with depth=2 in a random forest classifier for heart disease dataset

Comparing bagging, random forest and boosting

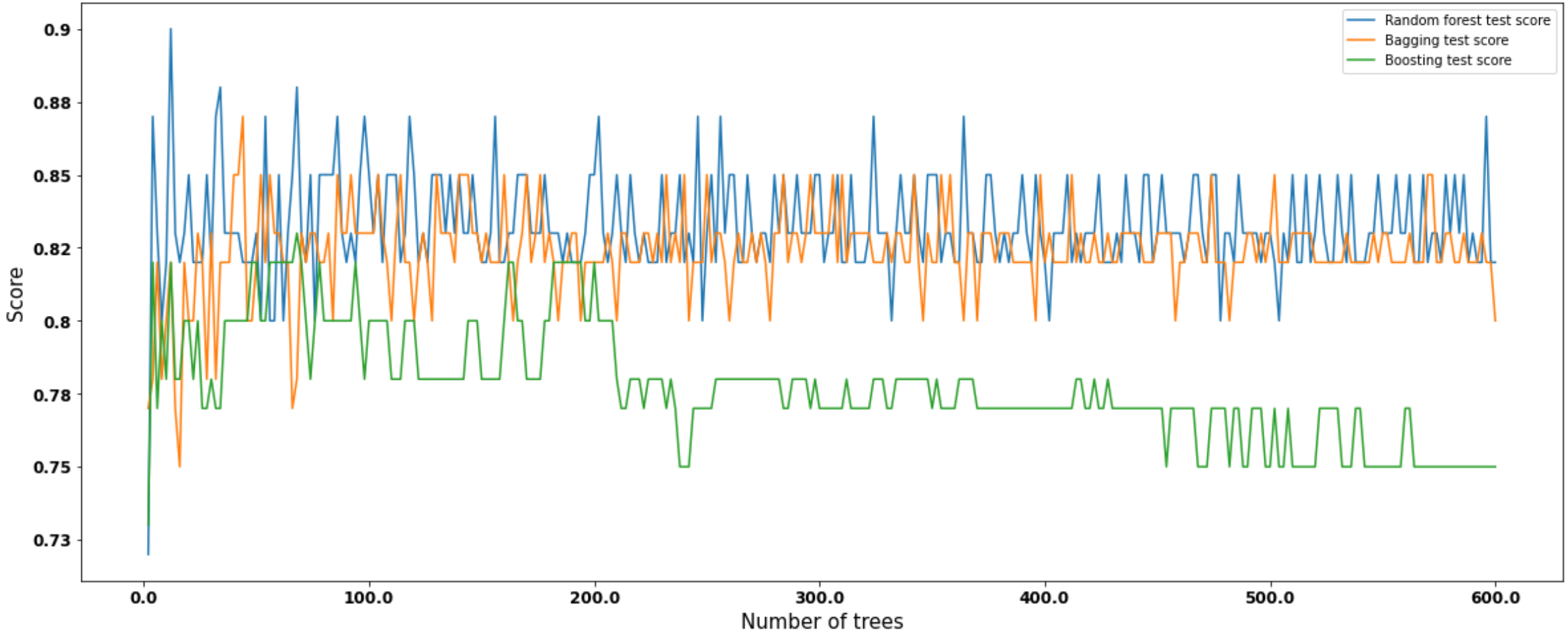
Classifier comparison: test set error

- **Dataset:** Heart data with features age, chest pain, etc., to predict the presence of heart disease
- **Source:** https://www.statlearning.com/resources-first-edition



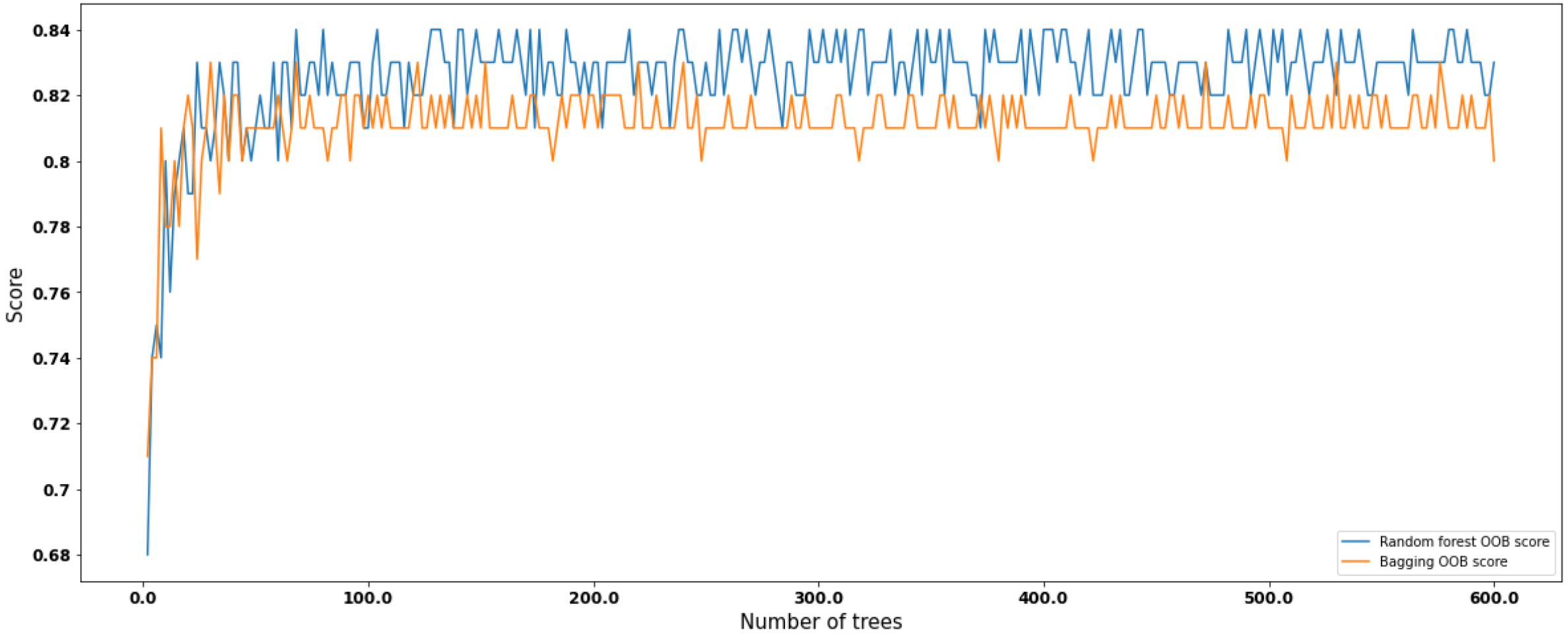
Classifier comparison: test set score

- **Dataset:** Heart data with features age, chest pain, etc., to predict the presence of heart disease
- **Source:** https://www.statlearning.com/resources-first-edition



Classifier comparison: OOB score

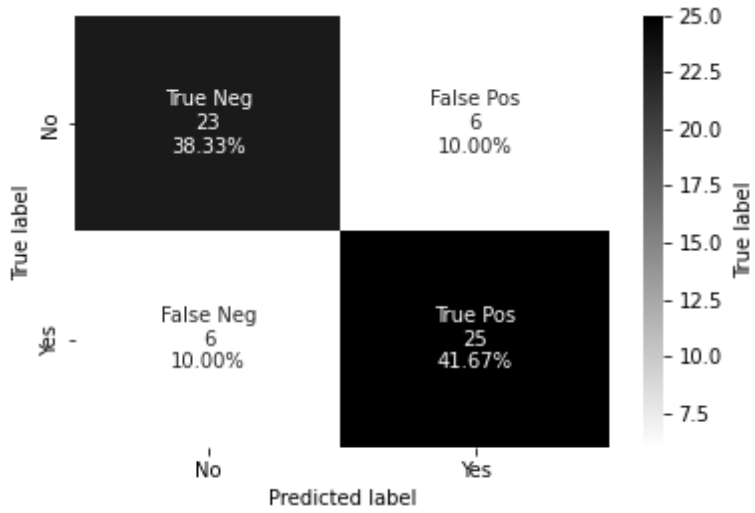
- **Dataset:** Heart data with features age, chest pain, etc., to predict the presence of heart disease
- **Source:** https://www.statlearning.com/resources-first-edition



Heart data: classifier comparison

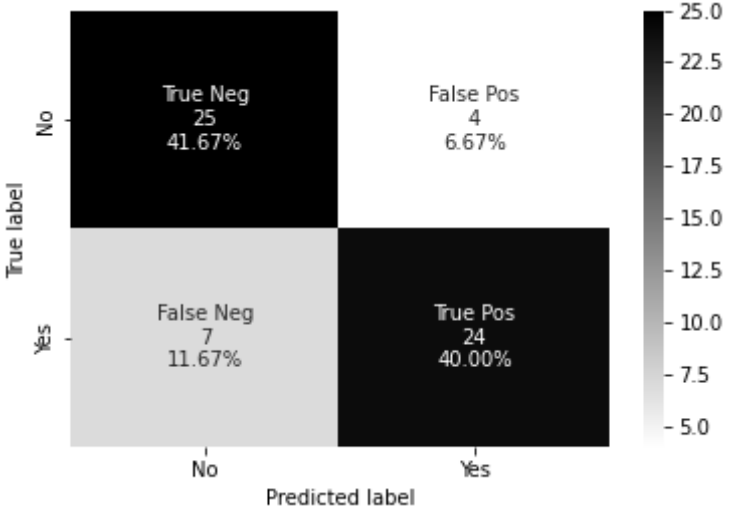
Bagging

Accuracy	0.8
Precision	0.81
Recall	0.81
F1 score	0.81



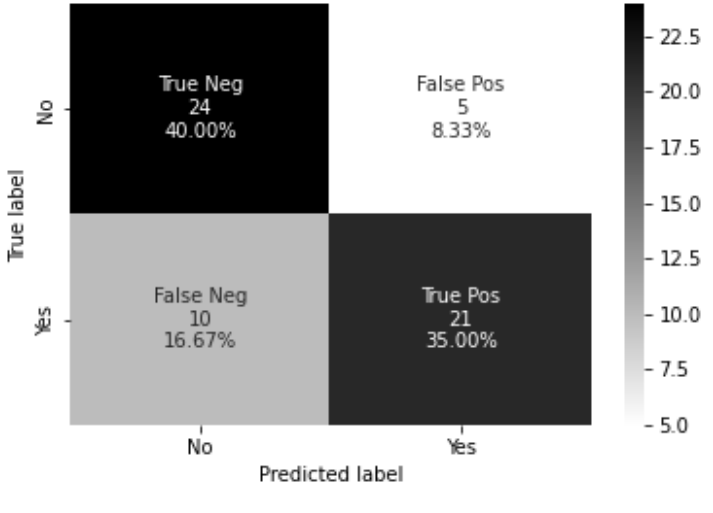
Random forest

Accuracy	0.82
Precision	0.86
Recall	0.77
F1 score	0.81



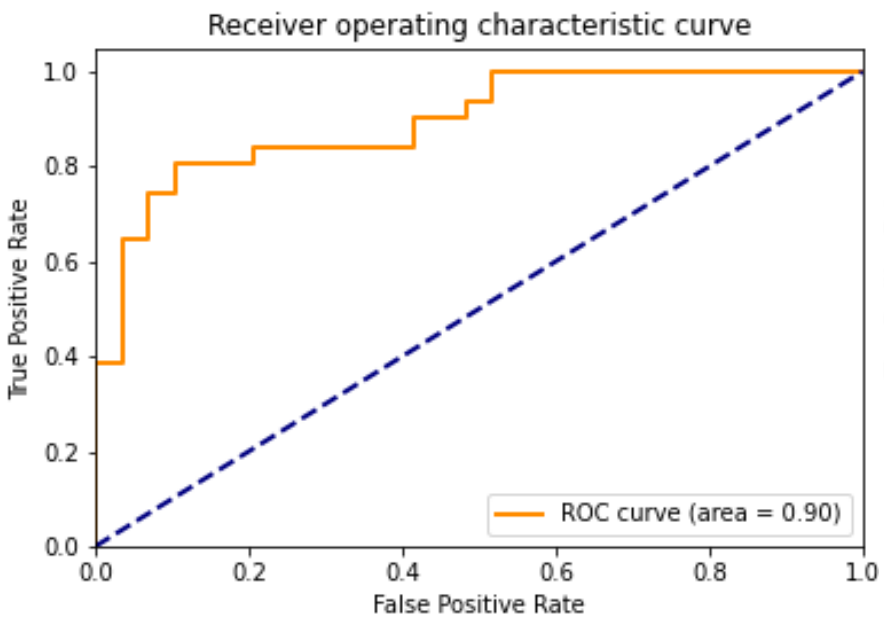
Boosting

Accuracy	0.75
Precision	0.81
Recall	0.68
F1 score	0.74

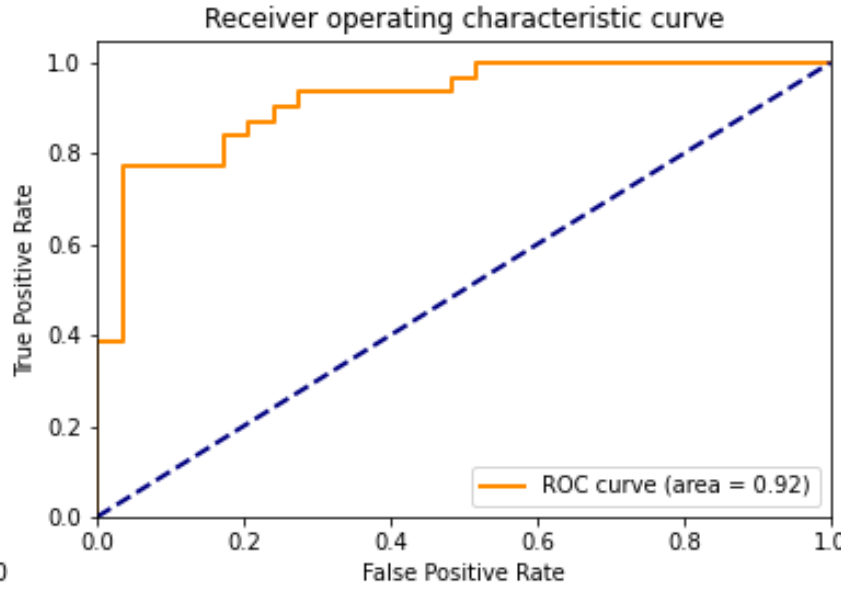


Heart data: classifier comparison

Bagging



Random forest



Boosting