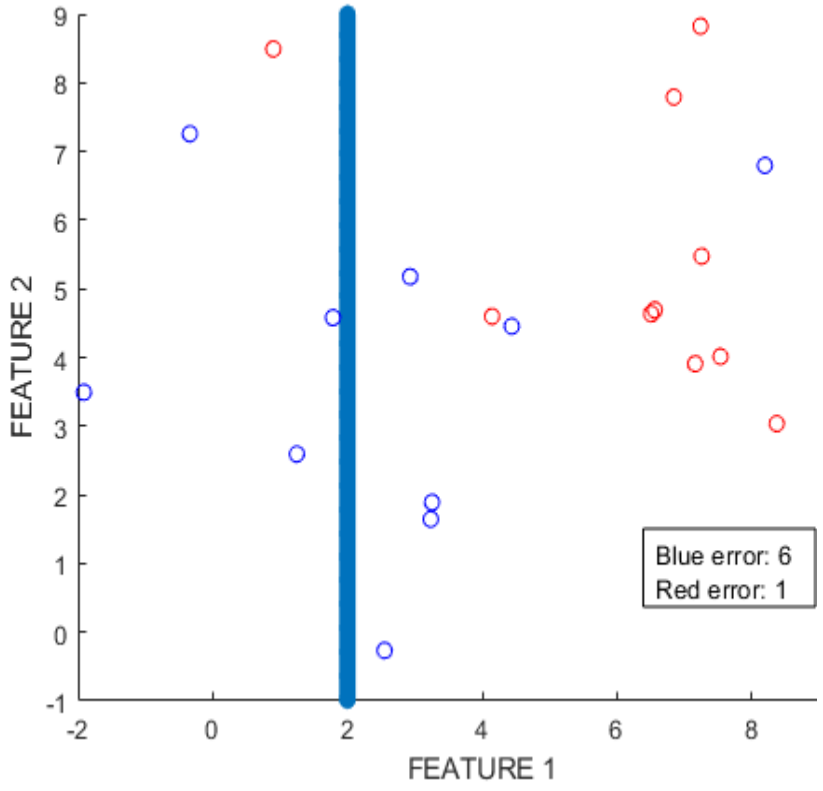
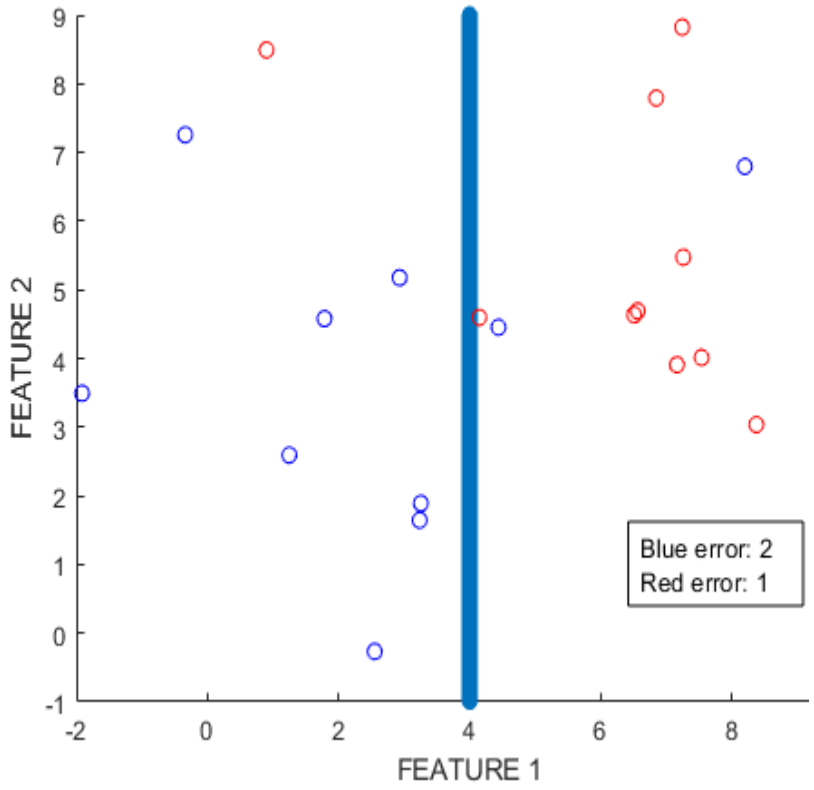
Perceptron and Multi-class Classification

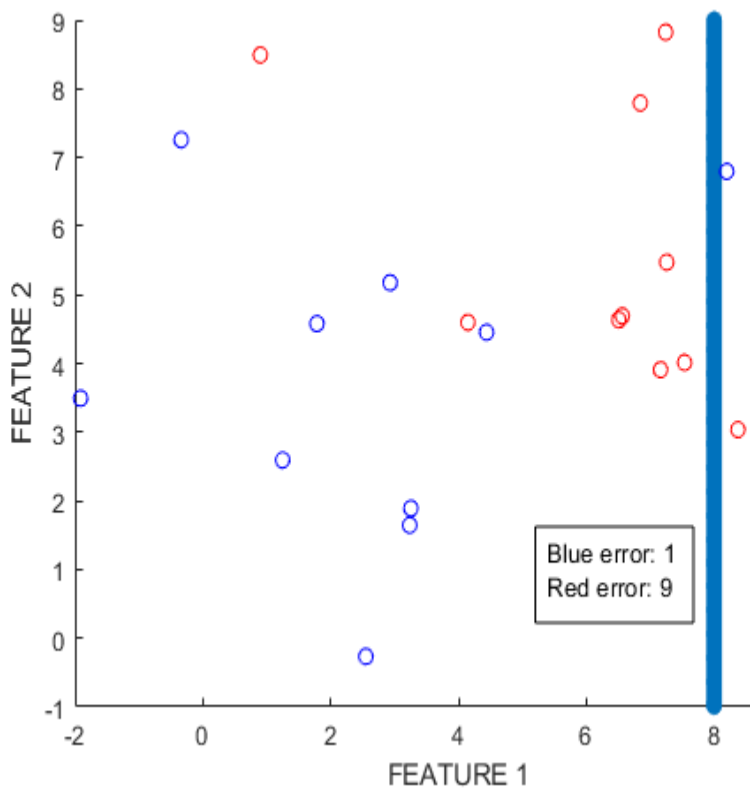
MACHINE LEARNING UNIT 15

Linear Classifiers

A line/hyperplane which separates the data-points from 2 classes

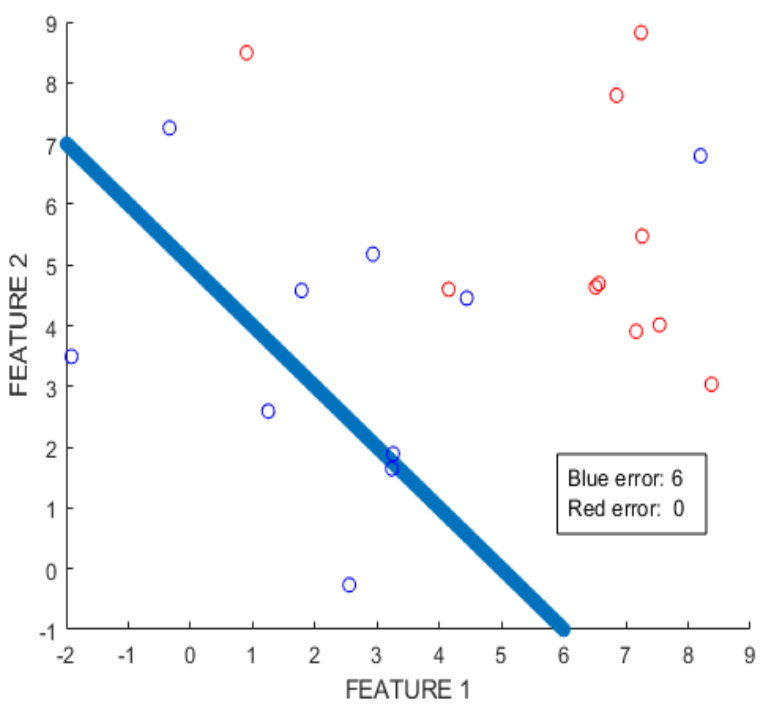


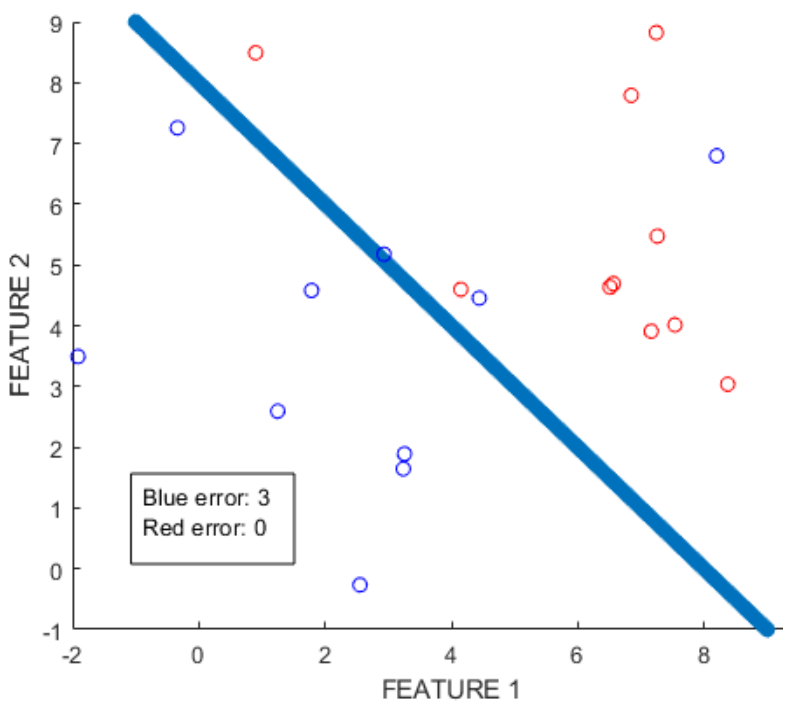


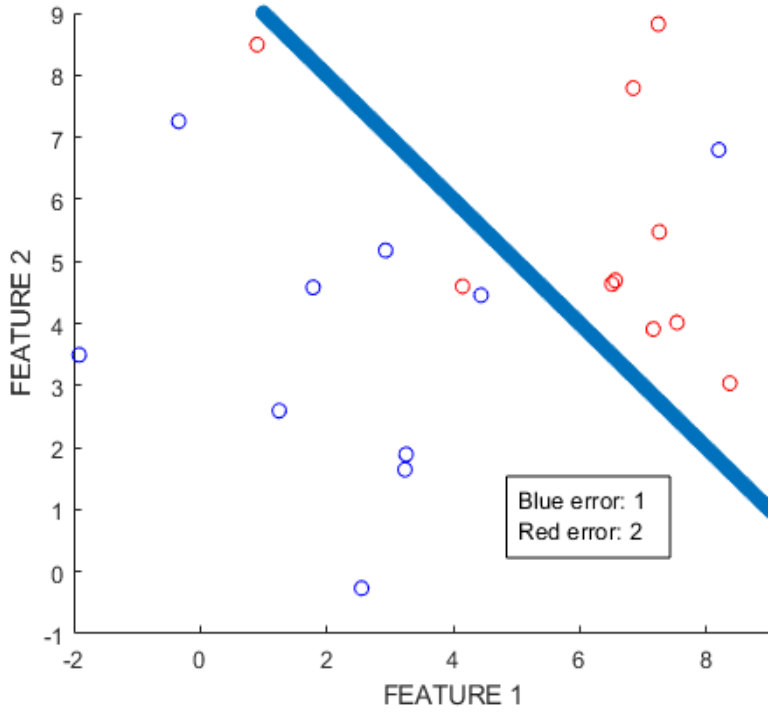


Linear Classifiers

A line/hyperplane which separates the data-points from 2 classes

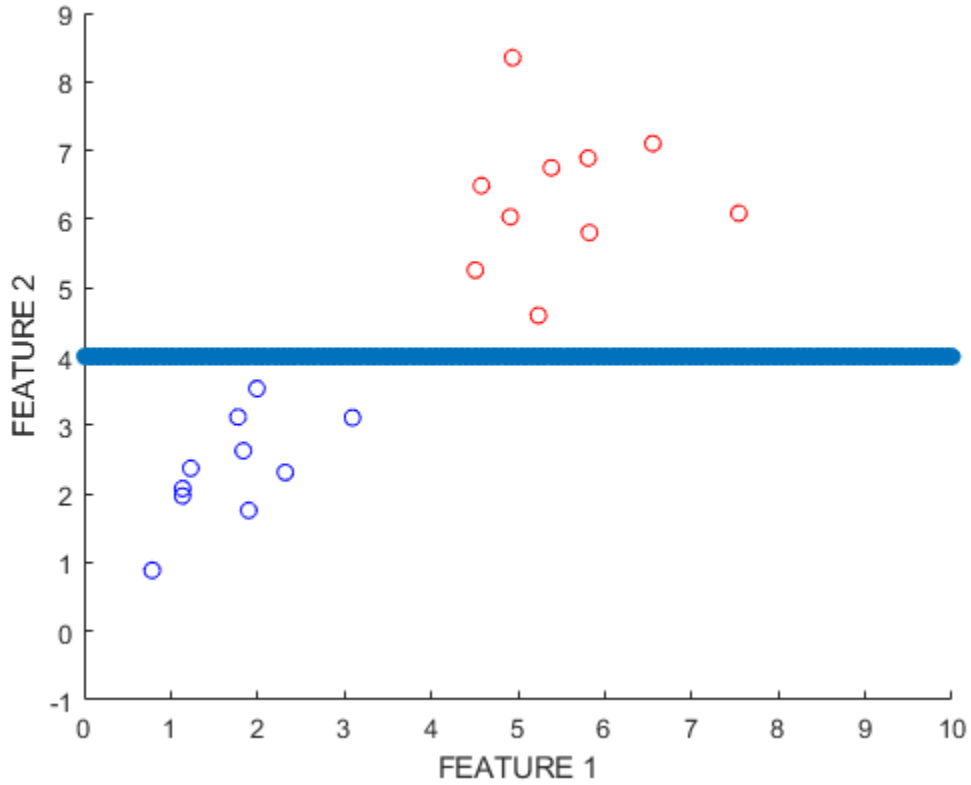


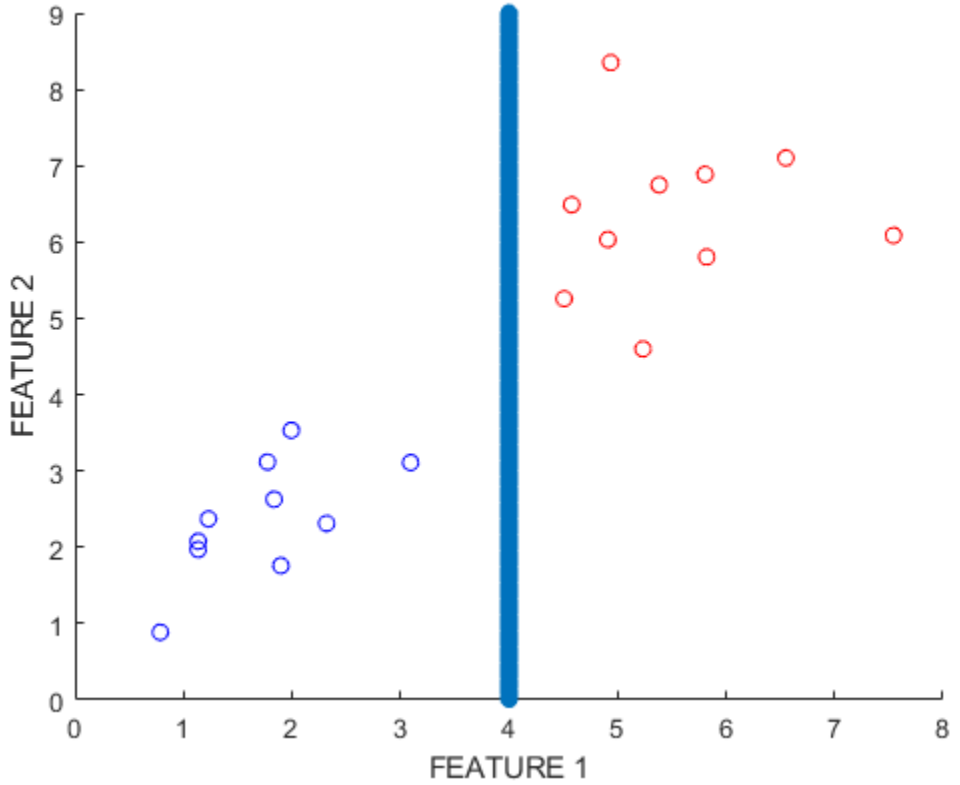




Linear Separability

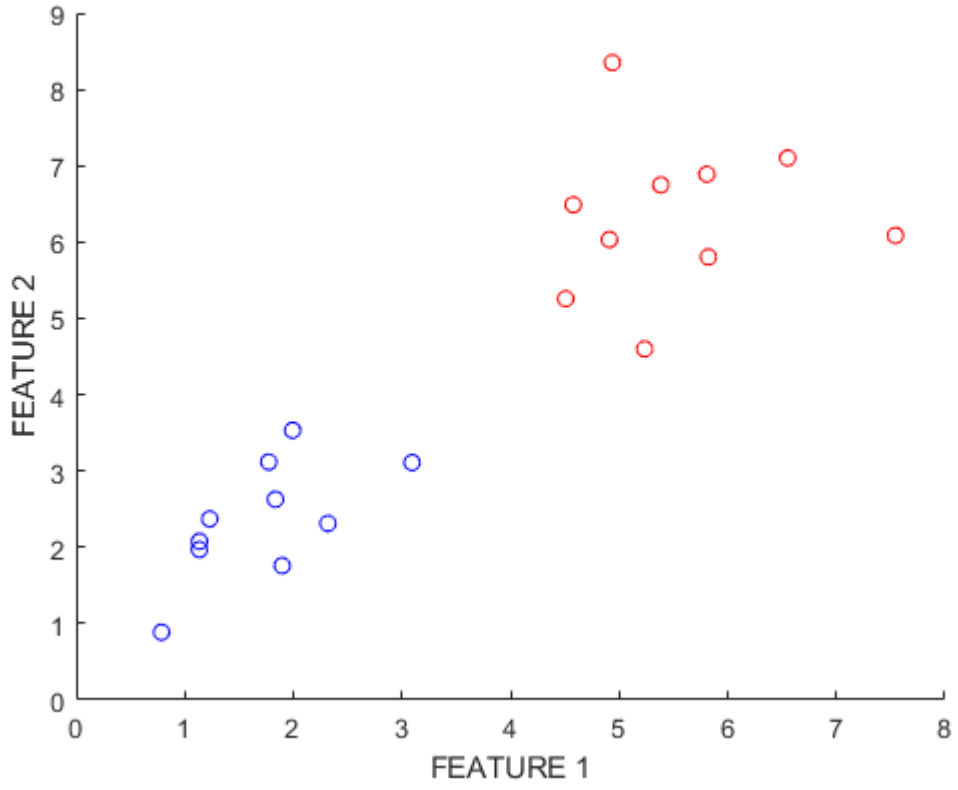
Linearly separable: A dataset for which there exists at least one linear classifier which perfectly separates the data-points into 2 classes



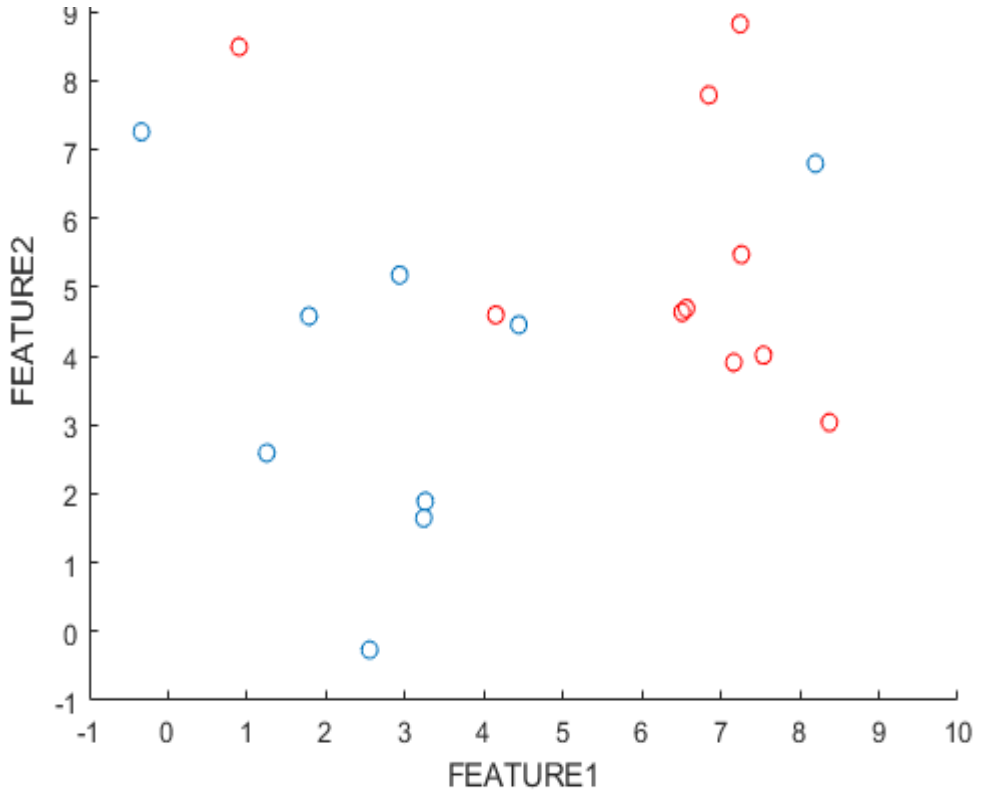


Linear Separability

Not all datasets are linearly separable



YES



NO

Linear Classifier

- Given a data-set, there may be any number (including 0) of linear classifiers, which **perfectly** separates the two
- But how can we find them?
- Perceptron algorithm can find one of them!
- Idea: 1) Start with an estimate

  2) Take each data-point and see if it can be correctly predicted.

  If not, update the estimate.

  3) Stop when all data-points are correctly classified

Perceptron Algorithm

- Input: (X_1,y_1), (X_2,y_2), …….. , (X_N,y_N) where y_i = +1 or -1
- Initialize W = [0 0 …. 0], b = 0
- For t = 1 to max_iter (or, till convergence)

    For i = 1 to N

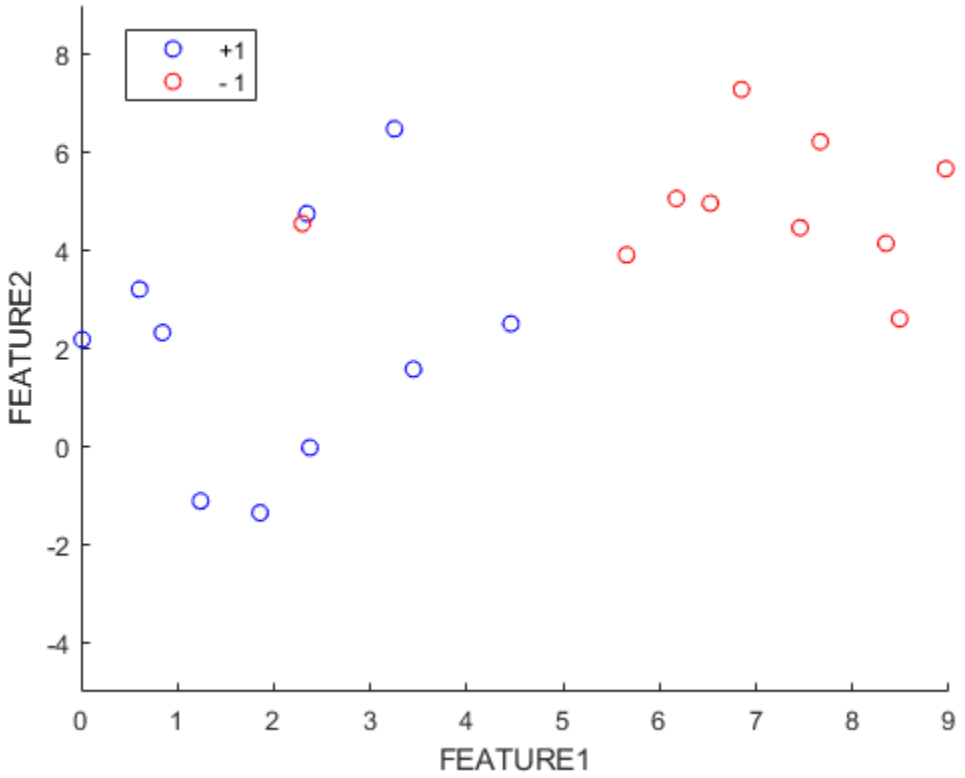
        if (y_i*sign(W.X_i + b)<0) ////misclassification!

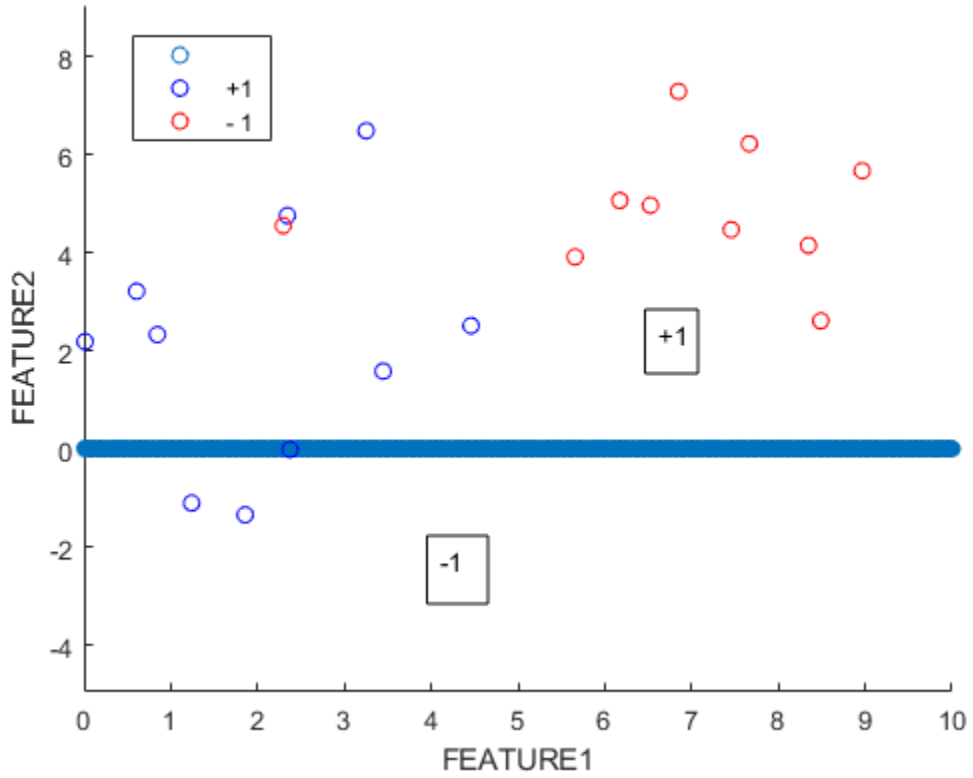
            W = W + y_iX_i ///update W

            b = b + y_i ///update b

Output: final values of (W, b)

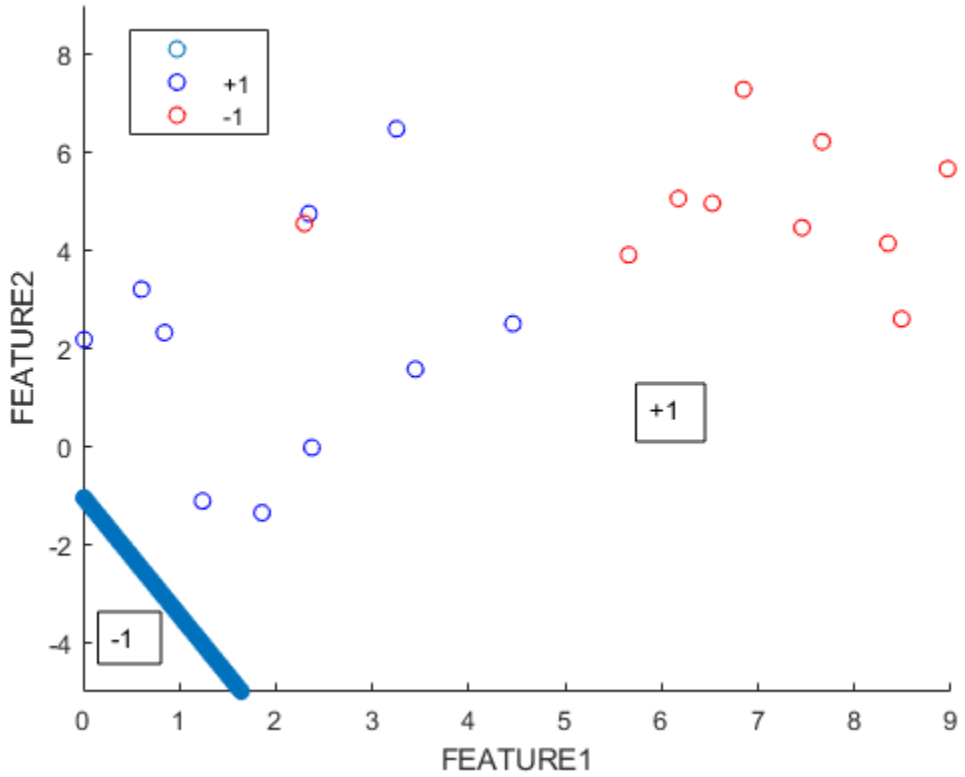
Illustration: Perceptron Algorithm



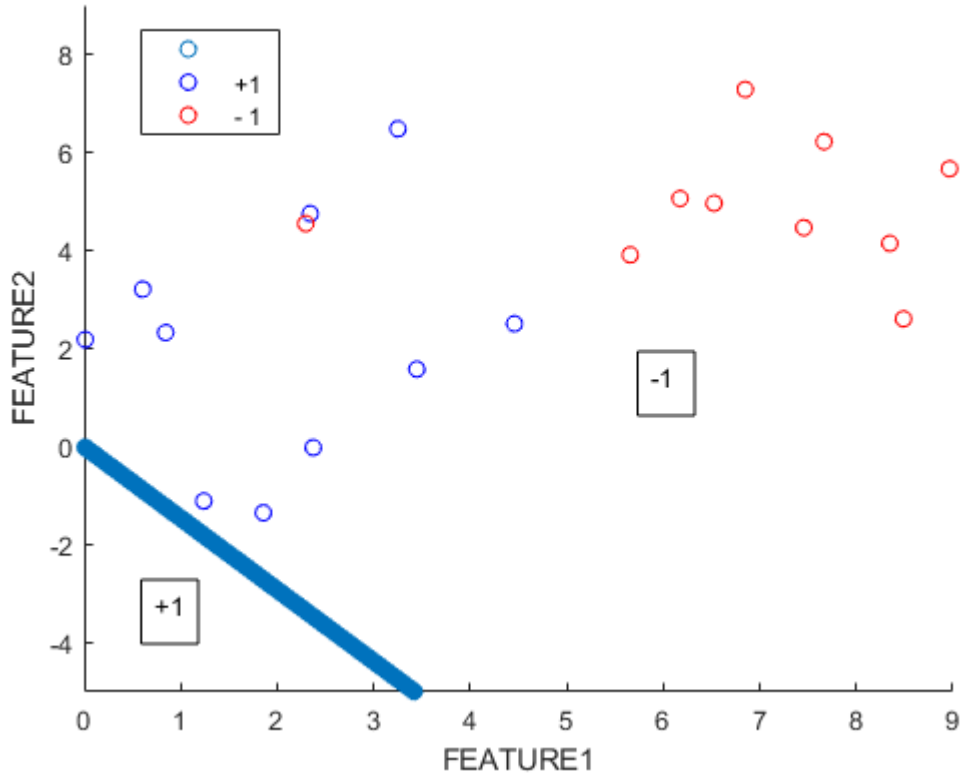


INITIALIZATION, ERROR = 13

Illustration: Perceptron Algorithm

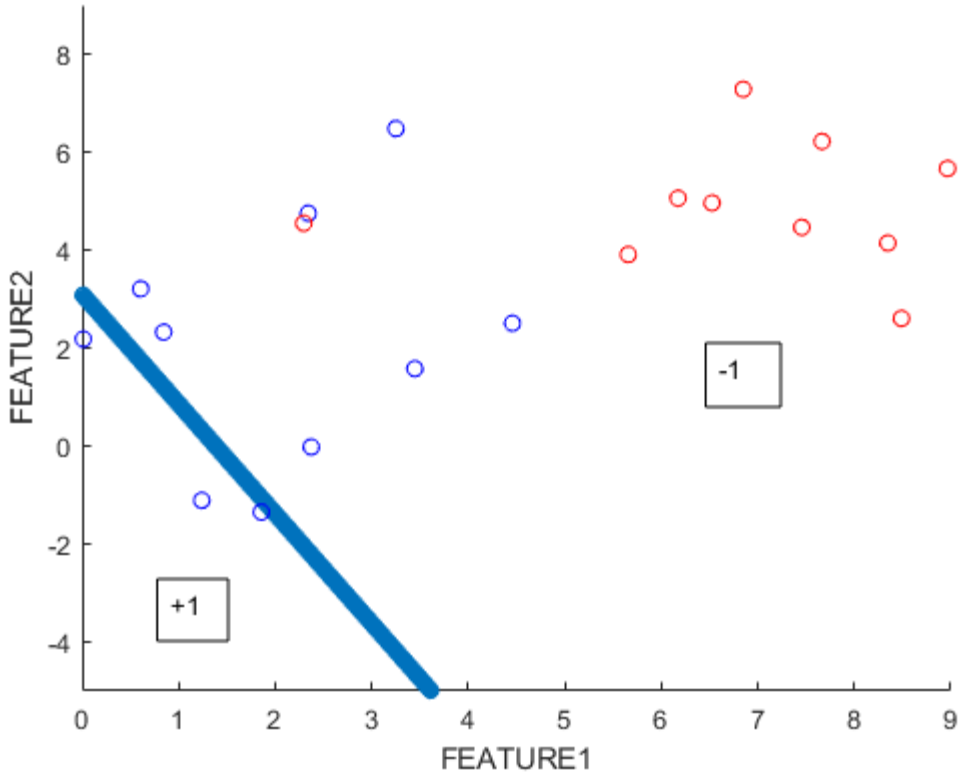


AFTER 10 POINTS, ERROR = 10

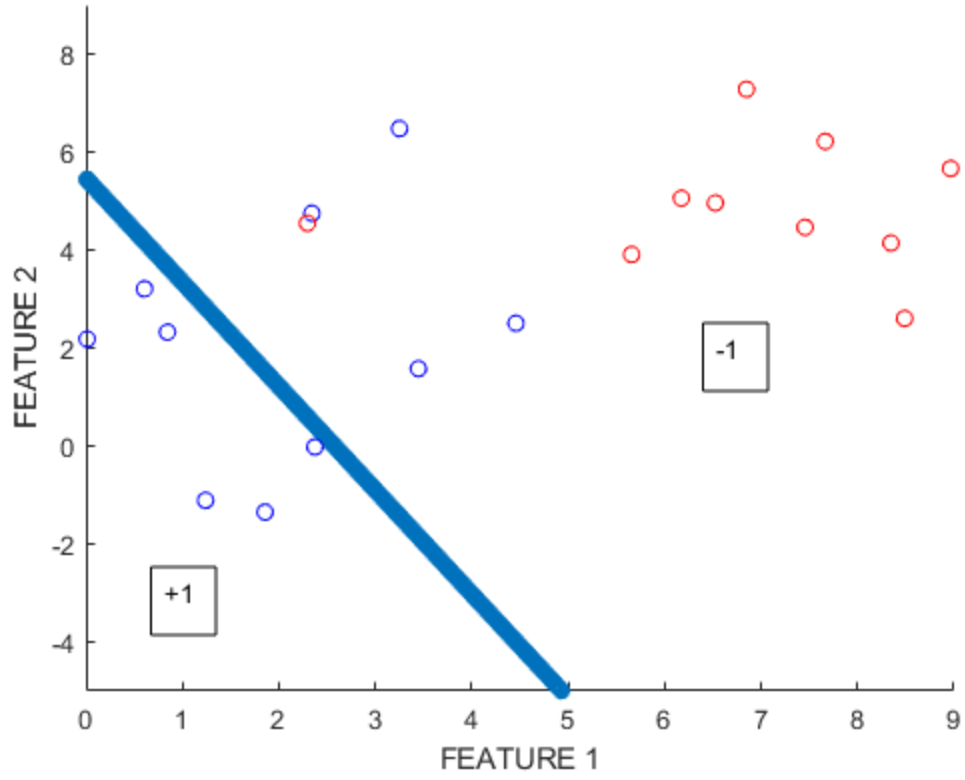


AFTER 1 ITERATION, ERROR = 10

Illustration: Perceptron Algorithm



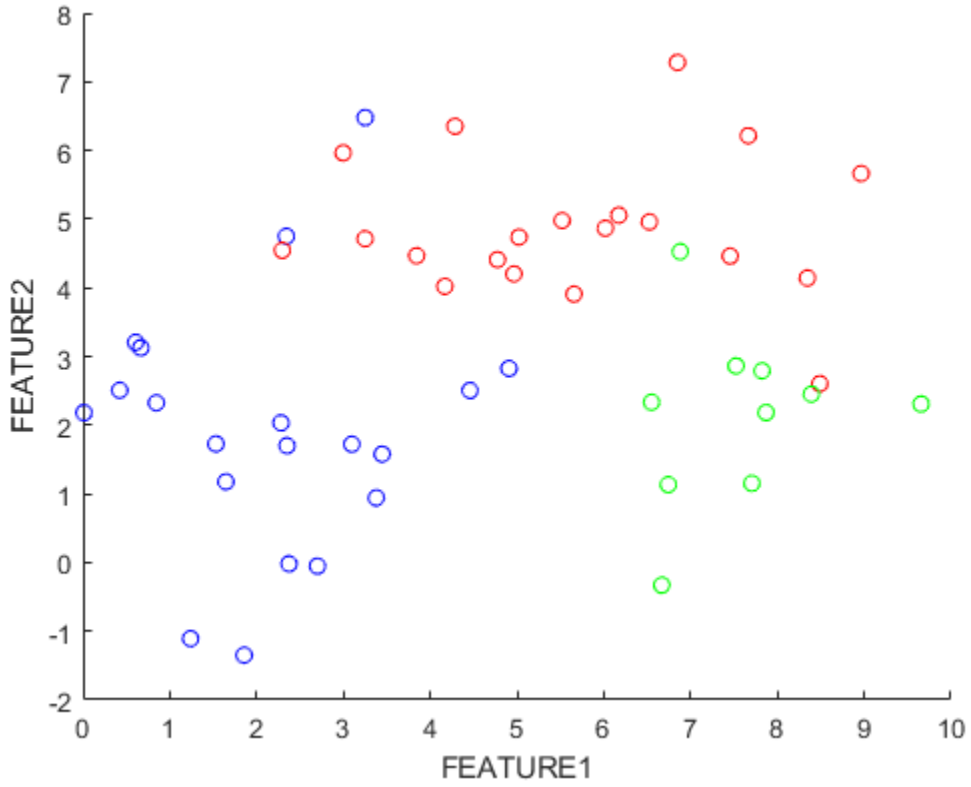
AFTER 7 ITERATIONS, ERROR = 7



AFTER 50 ITERATIONS, ERROR = 4

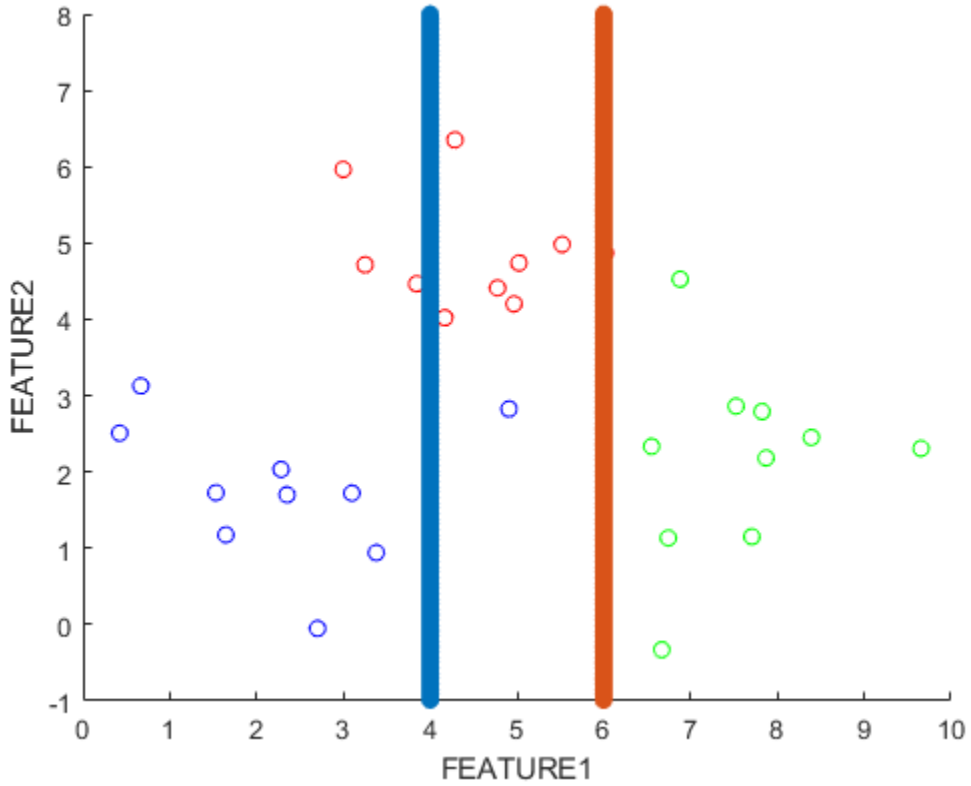
Multi-label classification

- A linear classifier creates 2 half-spaces
- So, it can do only binary classification!
- What for multi-label datasets?



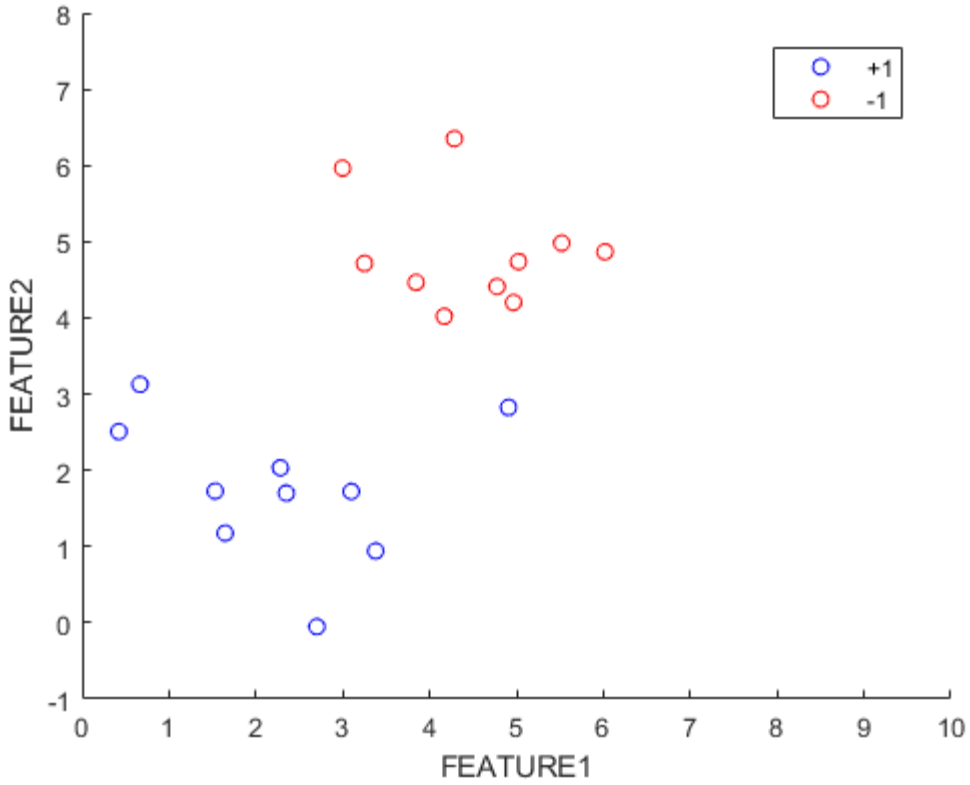
Multi-label classification

- A linear classifier creates 2 half-spaces
- So, it can do only binary classification!
- What for multi-label datasets?
- Multiple linear classifiers are needed!
- But how to find them?



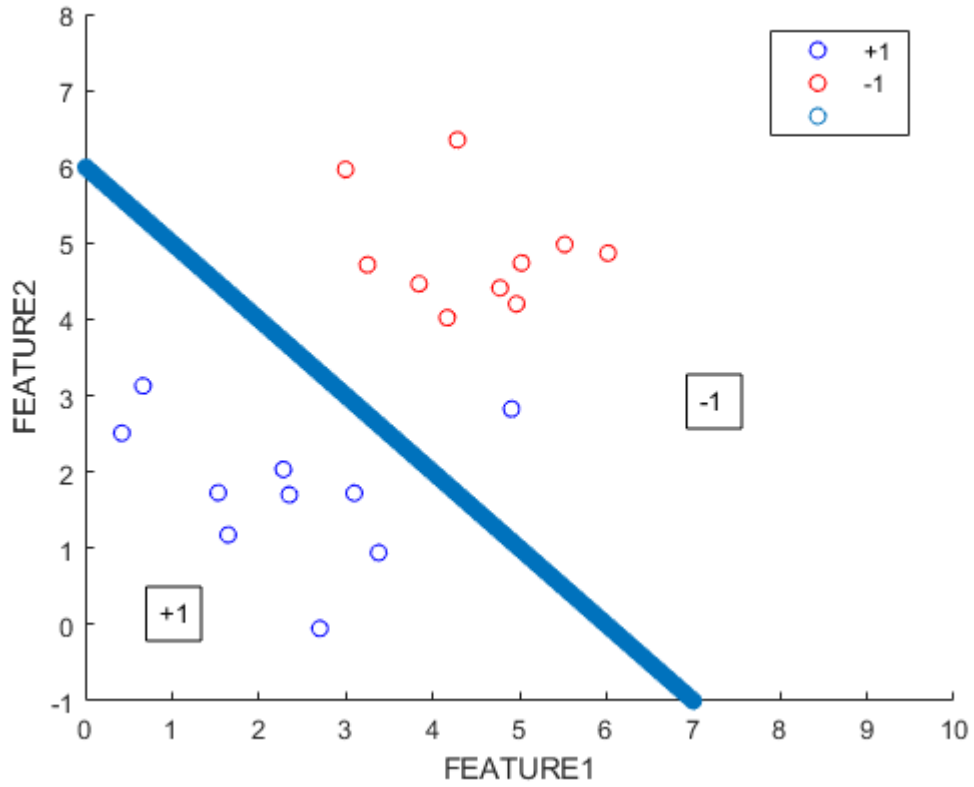
One-vs-one

- Take any two classes, disregard the rest
- Consider any of them as +1, other as -1



One-vs-one

- Take any two classes, disregard the rest
- Consider any of them as +1, other as -1
- Use perceptron algorithm to find linear classifier 1!

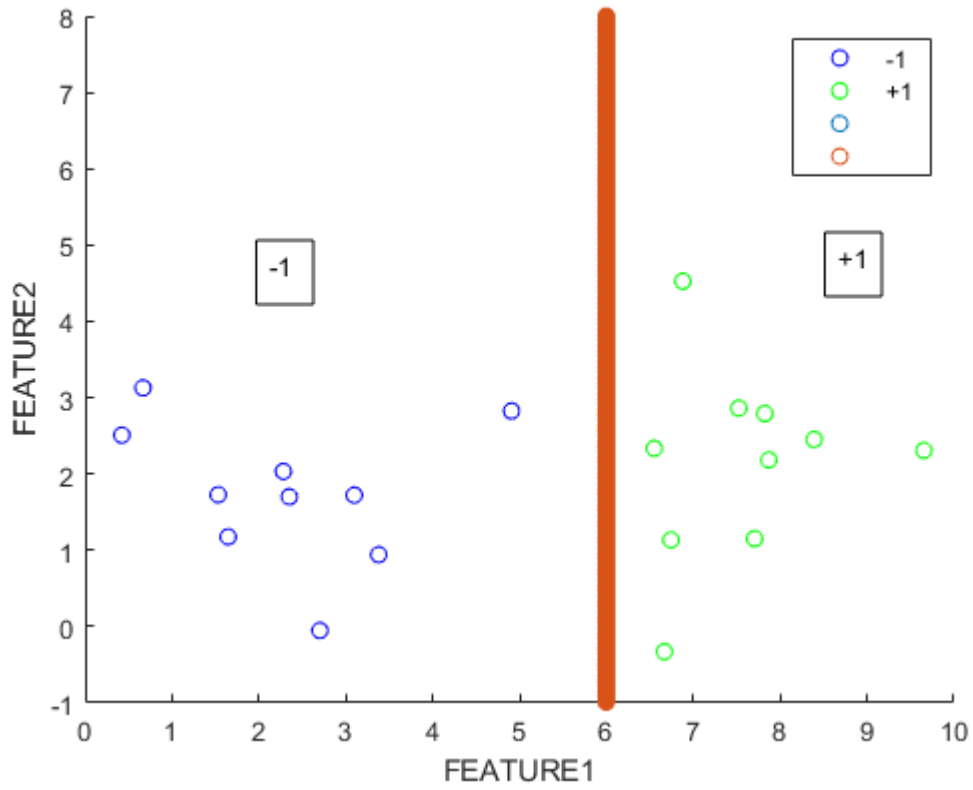


One-vs-one

- Take any two classes, disregard the rest
- Consider any of them as +1, other as -1
- Use perceptron algorithm to find

linear classifier 1!

- Repeat for other pairs of classes!

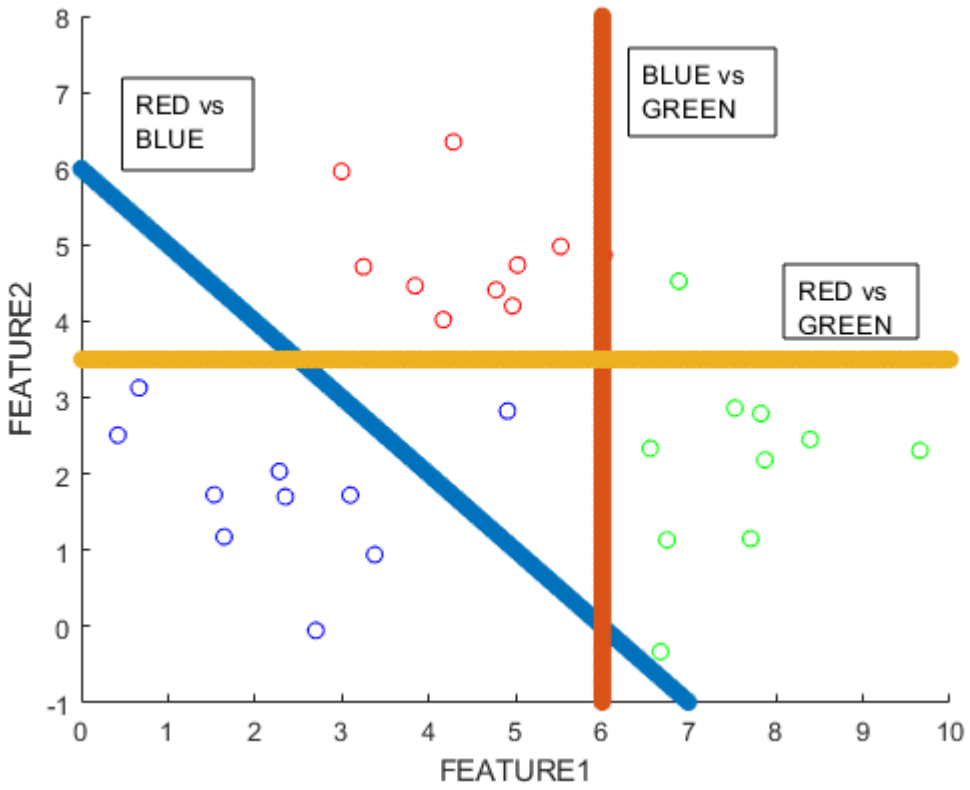


One-vs-one

- Take any two classes, disregard the rest
- Consider any of them as +1, other as -1
- Use perceptron algorithm to find

linear classifier 1!

- Repeat for other pairs of classes!
- Finally, get an “Ensemble” of (K-choose-2) classifiers!

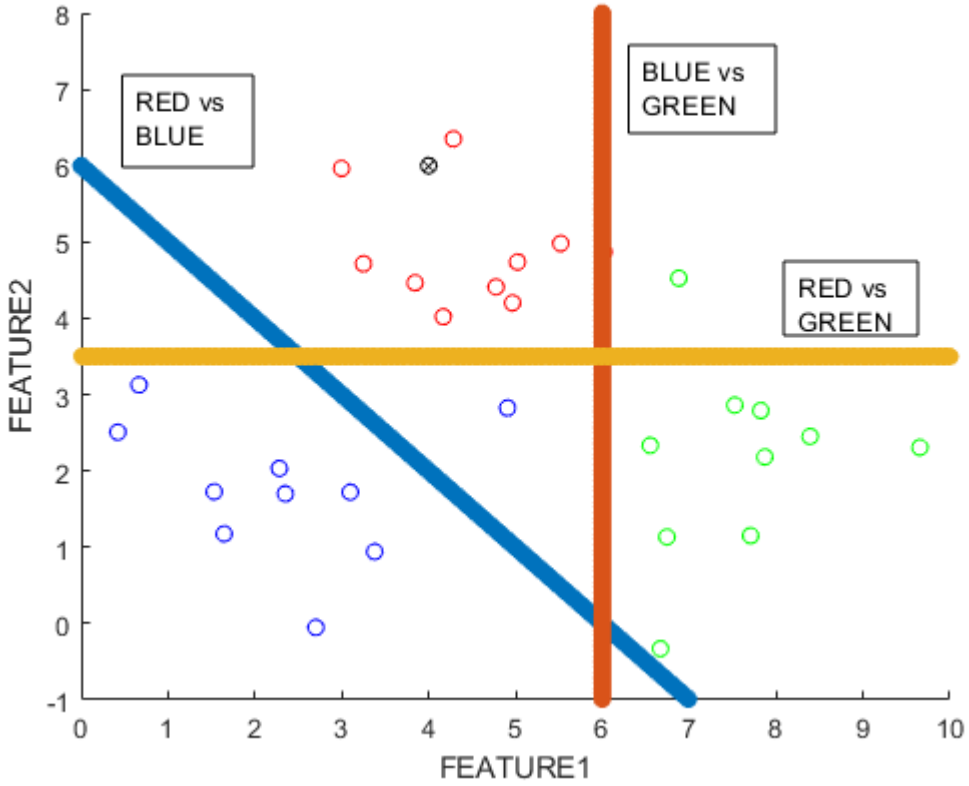


One-vs-one Classification

Given any new data-point, classify it with all one-vs-one classifiers!

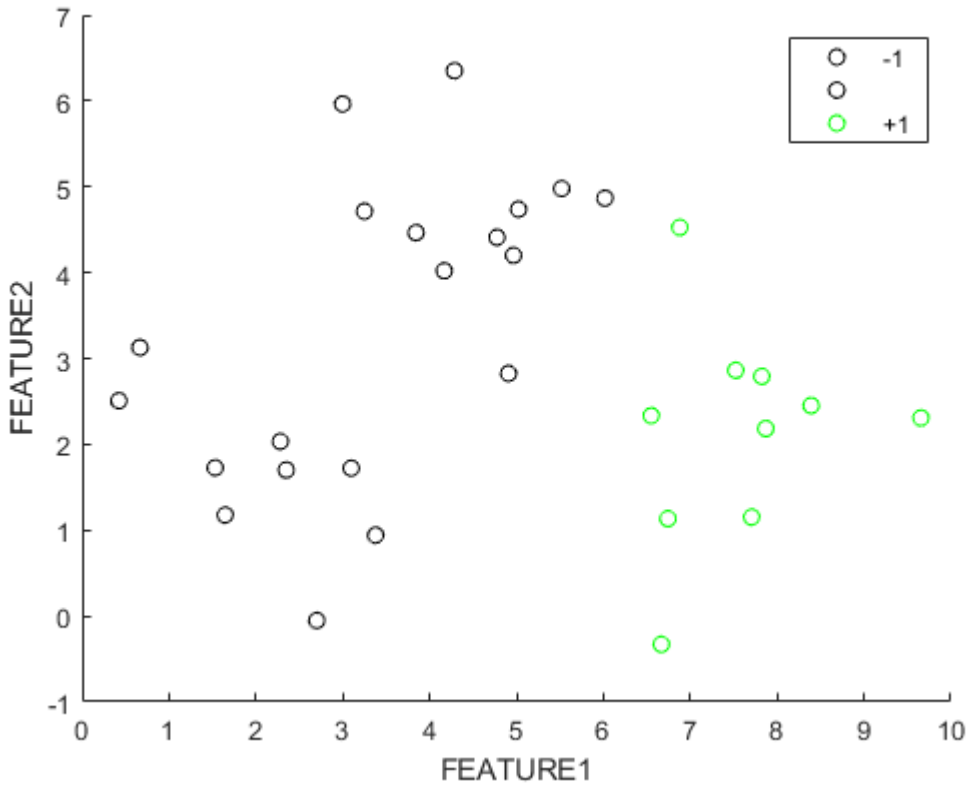
Take vote!

Classifier	Result	NET
RED vs BLUE	RED	RED-1, BLUE-0, GREEN-0
BLUE vs GREEN	BLUE	RED-1, BLUE-1, GREEN-0
RED vs GREEN	RED	RED-2, BLUE-1, GREEN-0
FINAL	RED!	



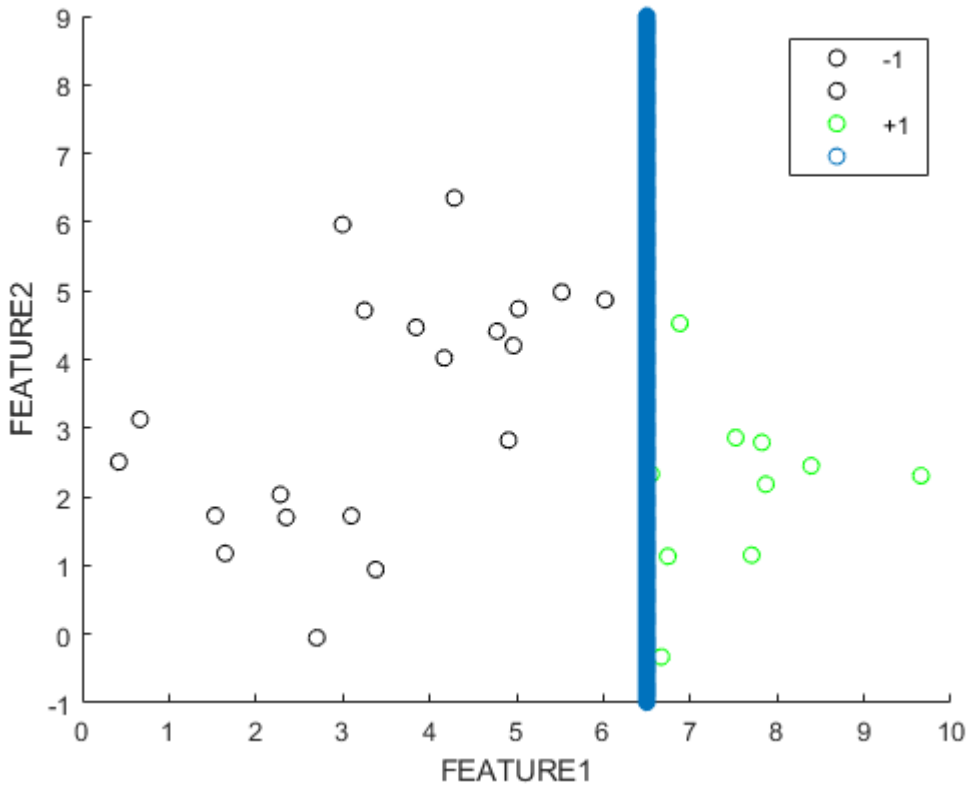
One-vs-all classification

- Take one class, consider it +1
- All other classes: -1



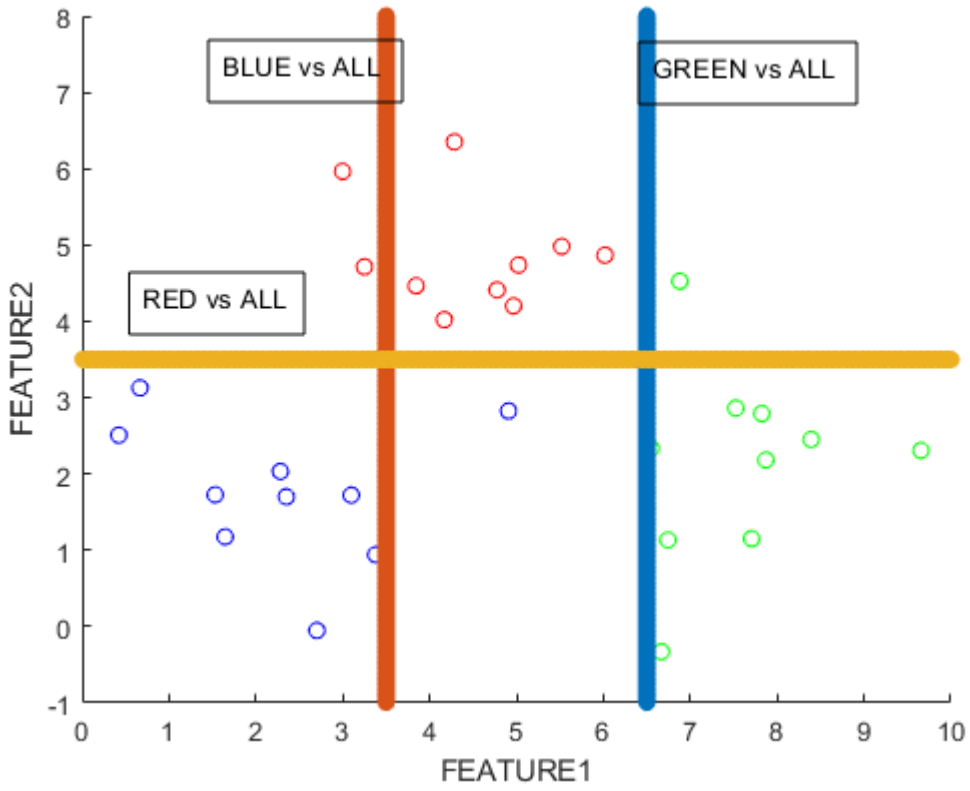
One-vs-all classification

- Take one class, consider it +1
- All other classes: -1
- Find linear classifier by perceptron!



One-vs-all classification

- Take one class, consider it +1
- All other classes: -1
- Find linear classifier by perceptron!
- Repeat for all classes, get ensemble



One-vs-all classification

Given any new data-point, classify it with all one-vs-one classifiers!

Take vote!

Classifier	Result	NET
RED vs ALL	RED	RED-1, BLUE-0, GREEN-0
BLUE vs ALL	ALL	RED-1, BLUE-0, GREEN-0
GREEN vs ALL	ALL	RED-1, BLUE-0, GREEN-0
FINAL	RED!