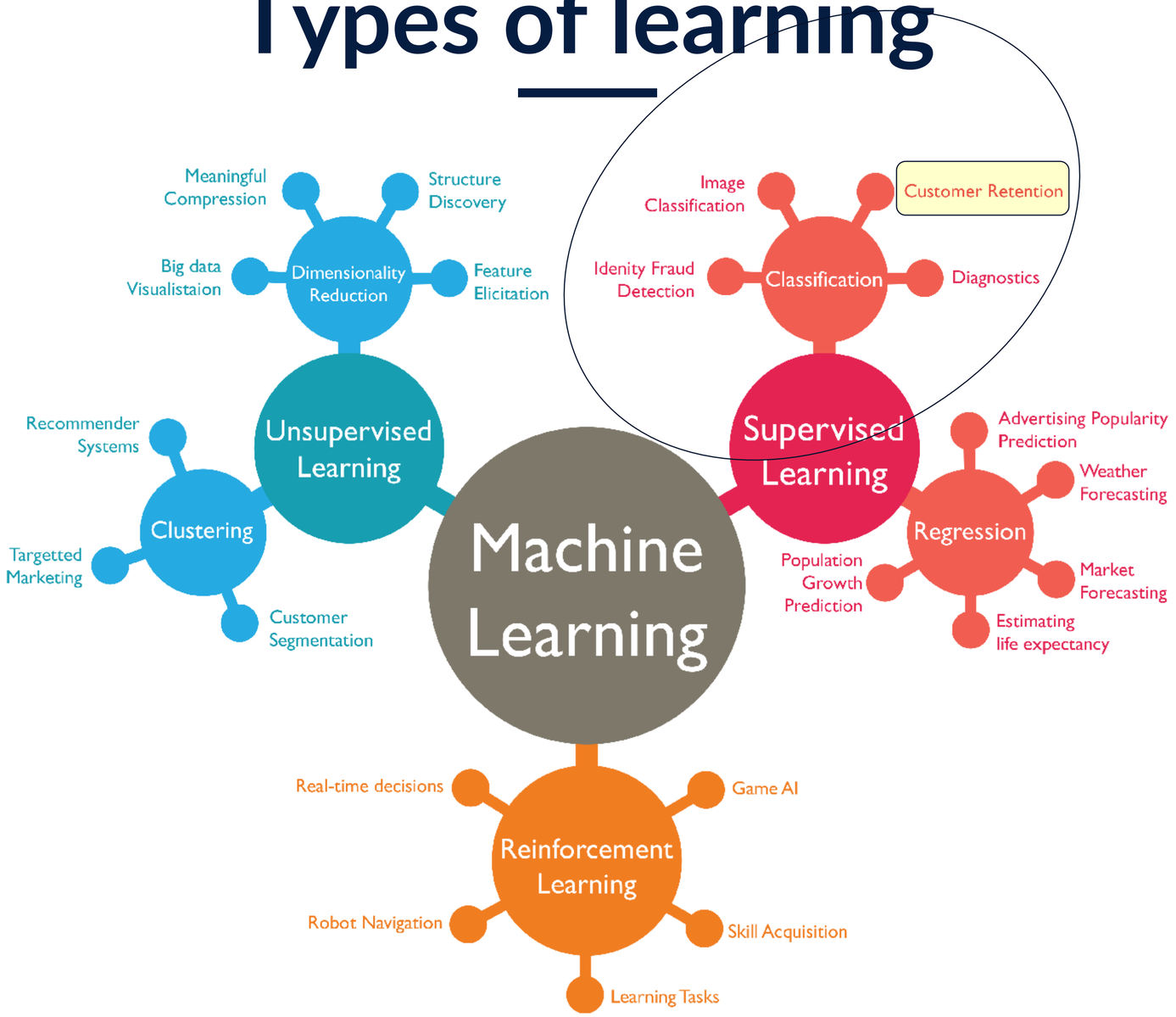
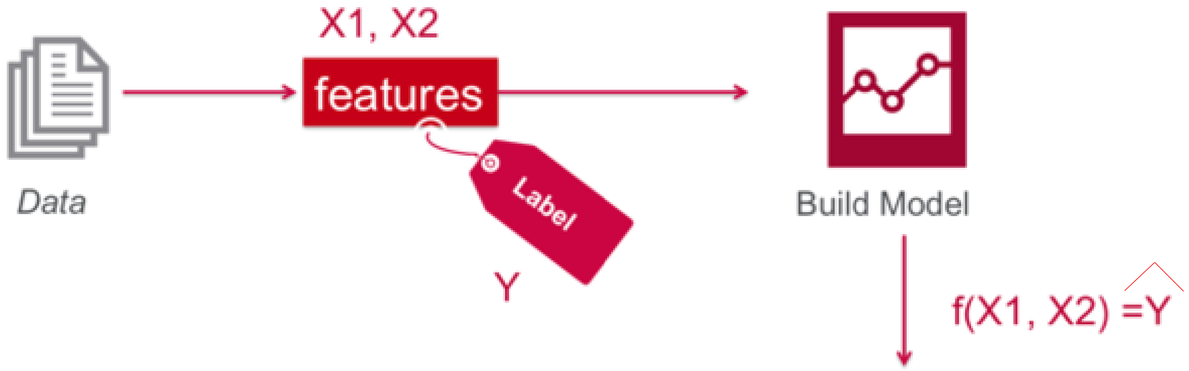
Linear classification

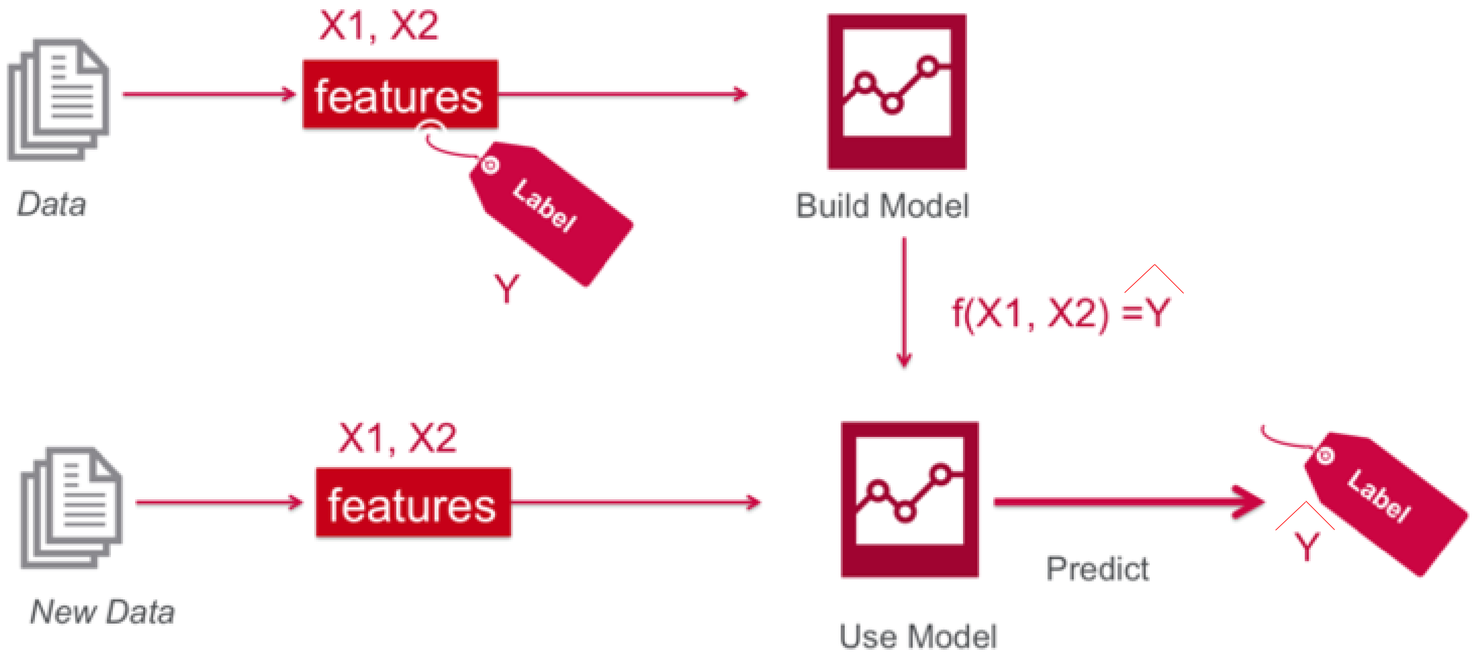
Types of learning



Supervised learning



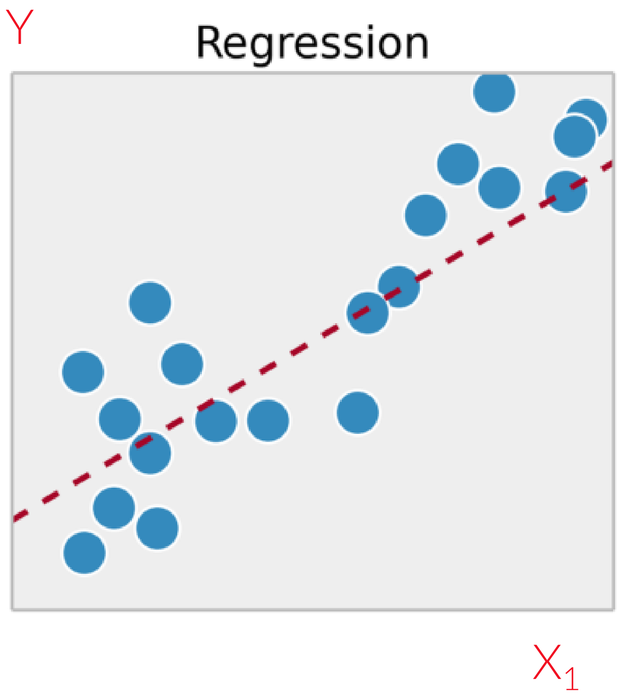
Supervised learning



Classification vs Regression

Classification: predict a discrete (or categorical) value based on the input variables

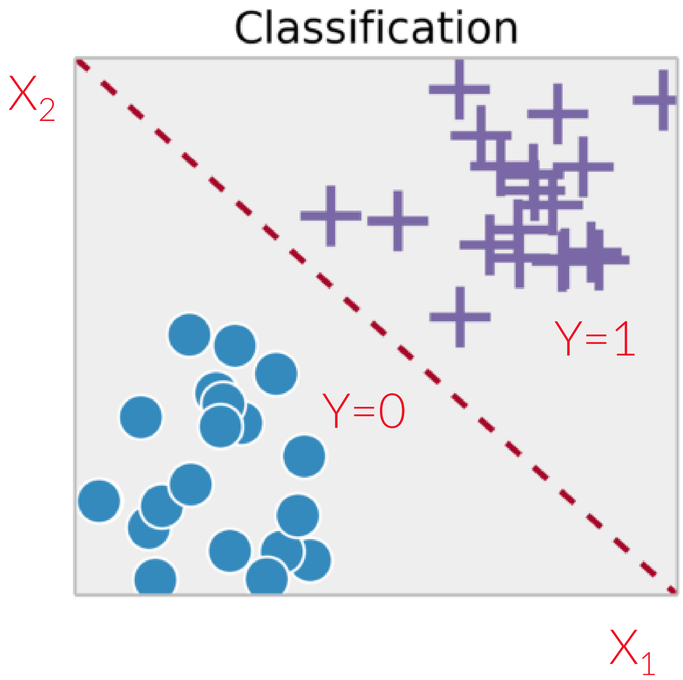
Regression: predict a continuous value based on the input variables



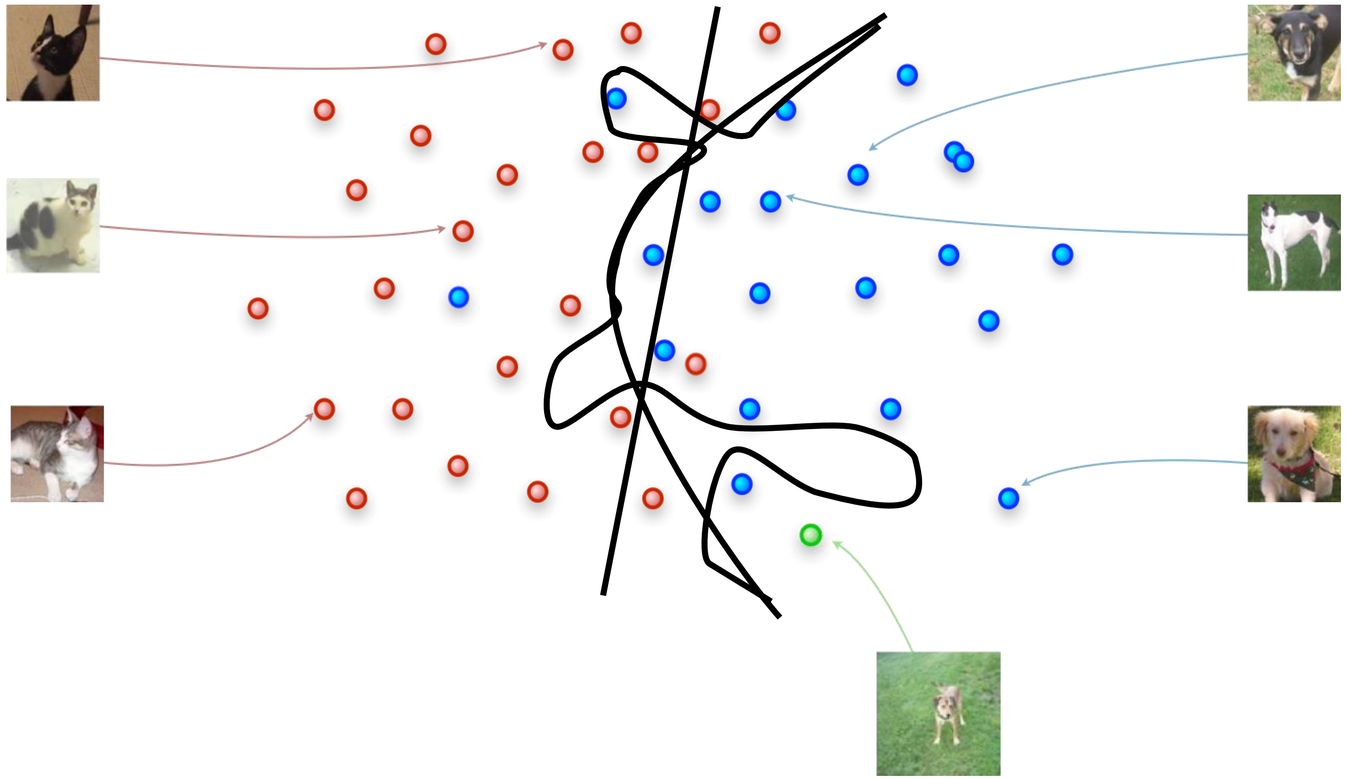
Classification vs Regression

Classification: predict a discrete (or categorical) value based on the input variables

Regression: predict a continuous value based on the input variables



Binary classification



Logistic regression

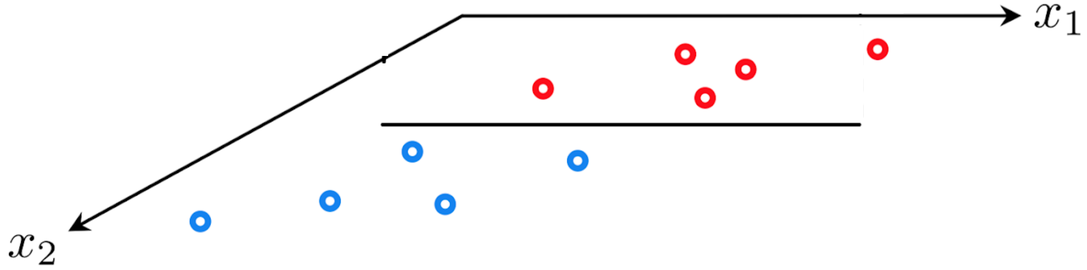
Linear classification

(Focus on binary)

Source: J. Watt et al., “Machine Learning refined”, Cambridge University Press

Two perspectives on classification

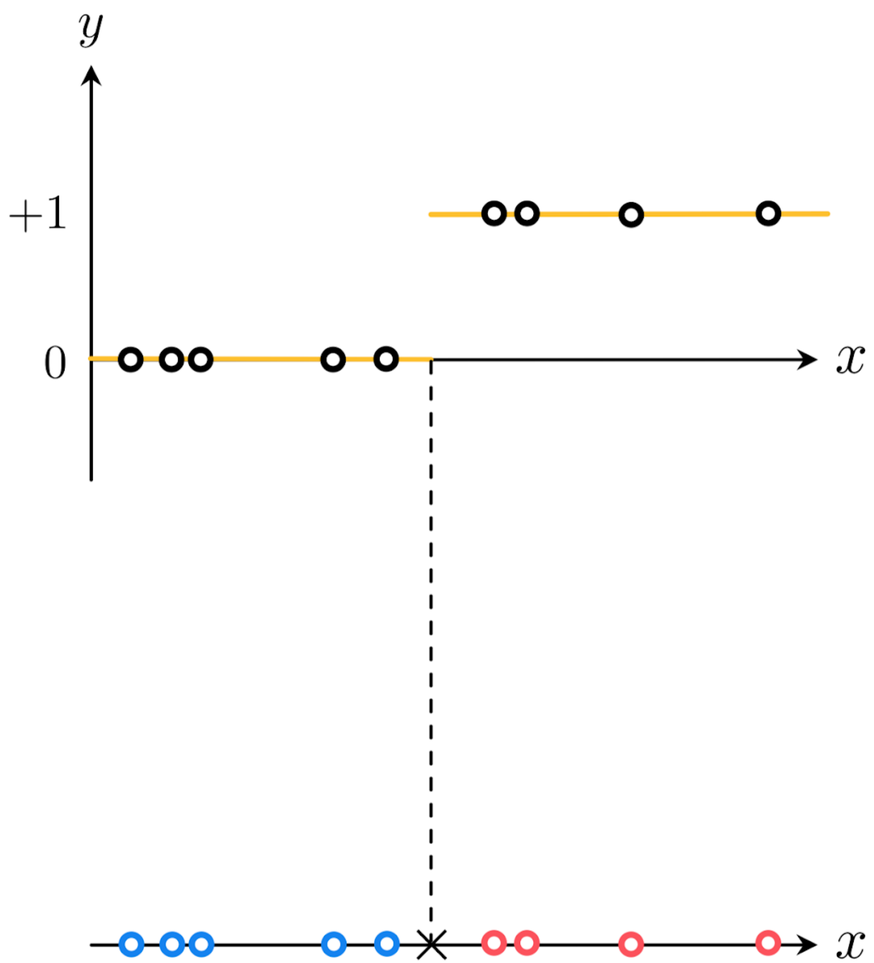
1d input: decision boundary is a point



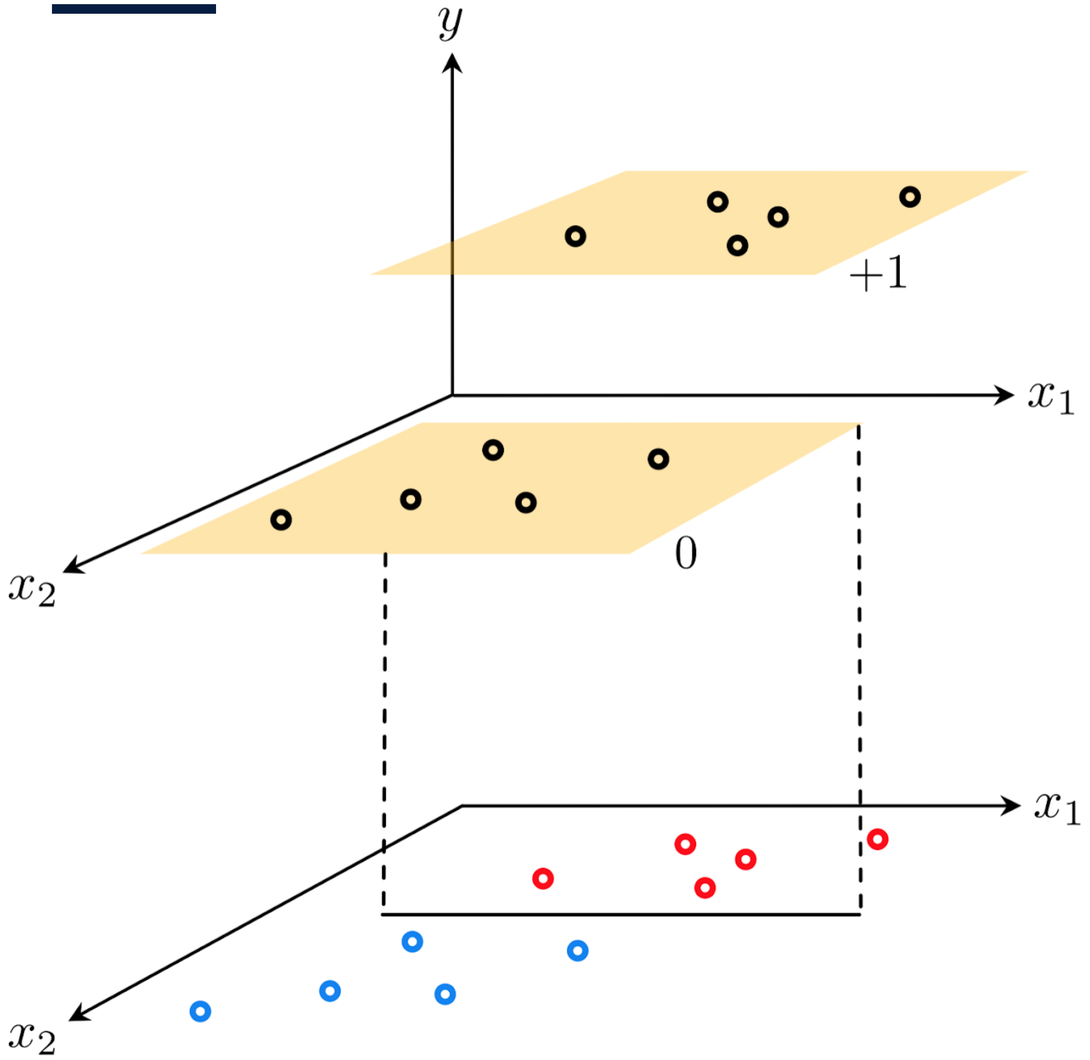
2d input: decision boundary is a line

Source: J. Watt et al., “Machine Learning refined”, Cambridge University Press

Two perspectives on classification

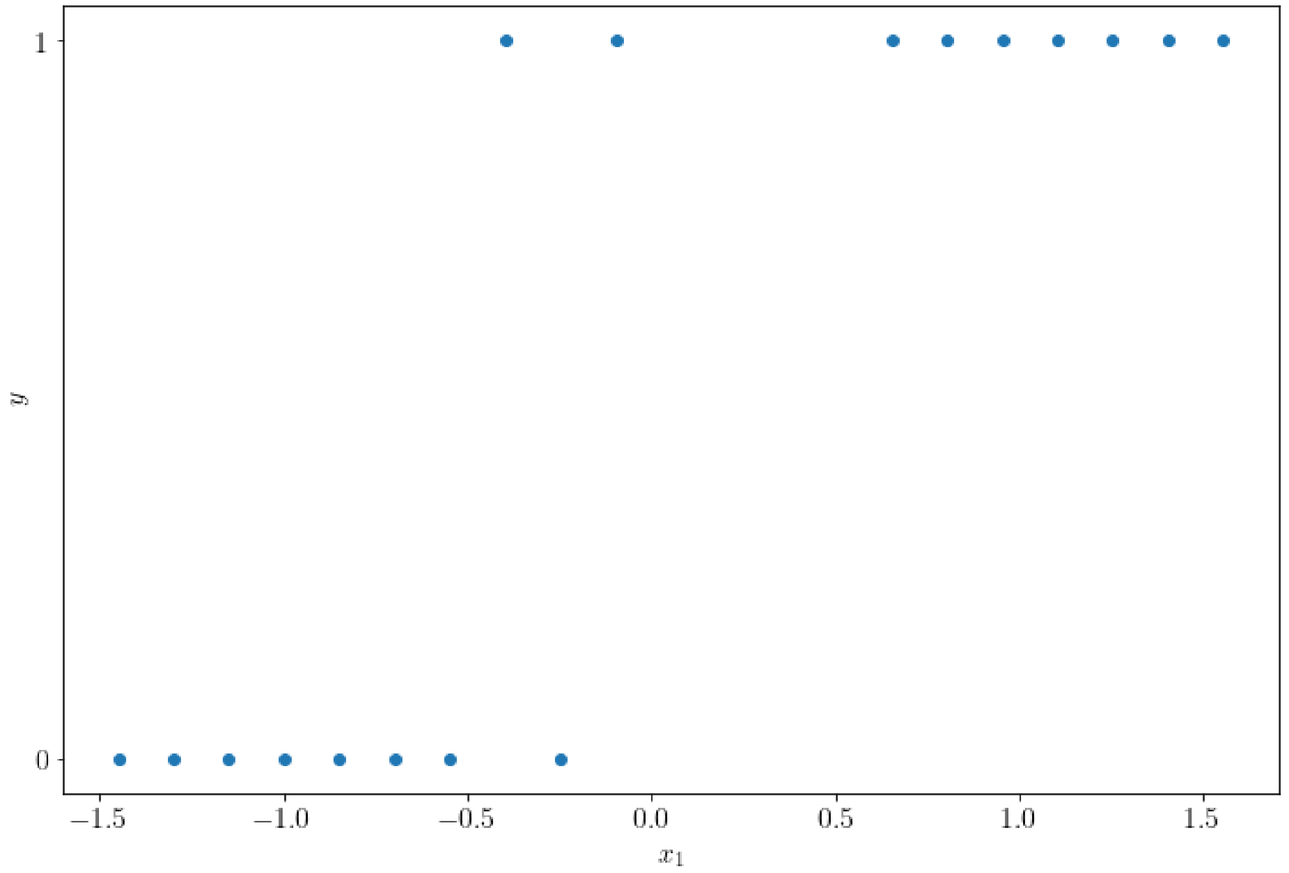


1d input: decision boundary is a point



2d input: decision boundary is a line

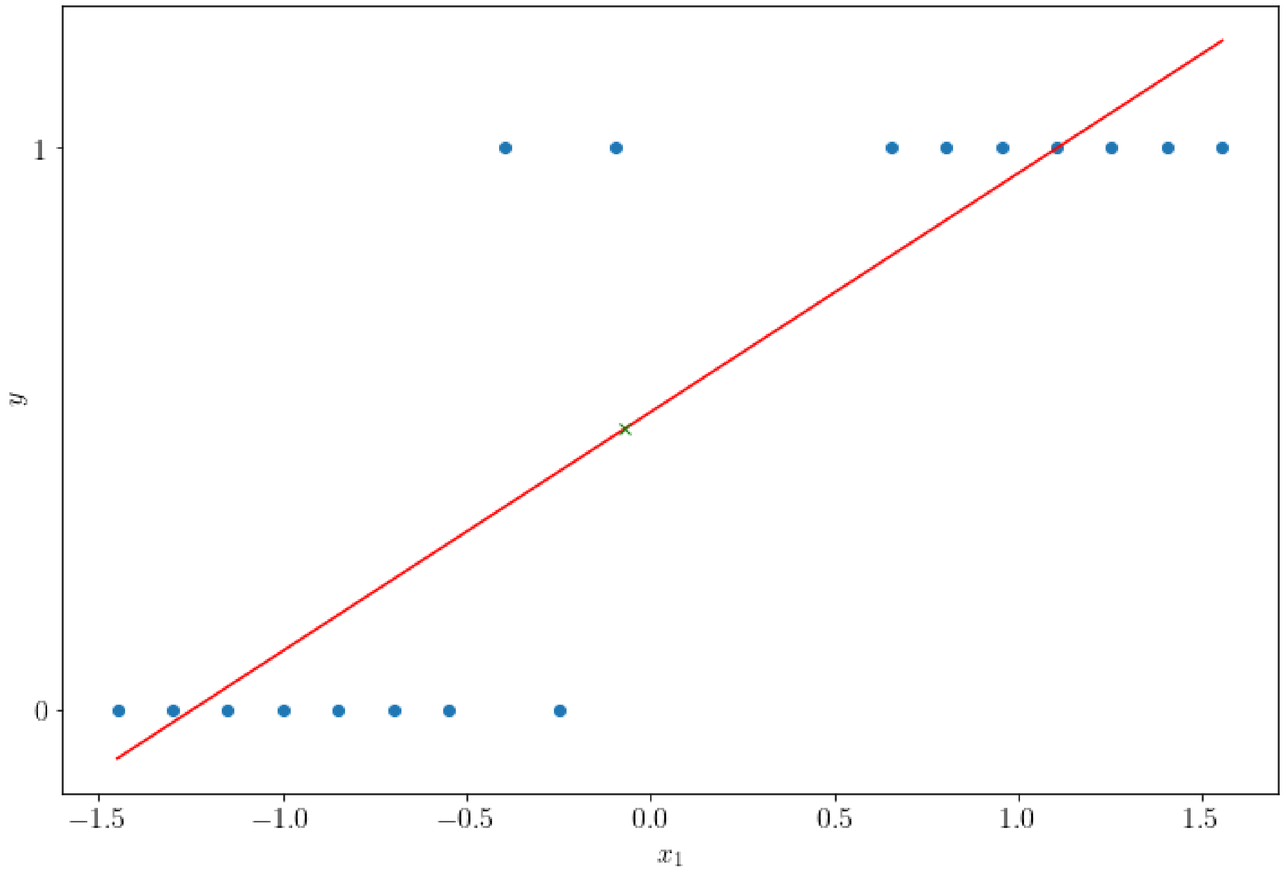
Illustrative 1D example



Proposition #1

How about applying the linear regression framework seen earlier to fit such data?

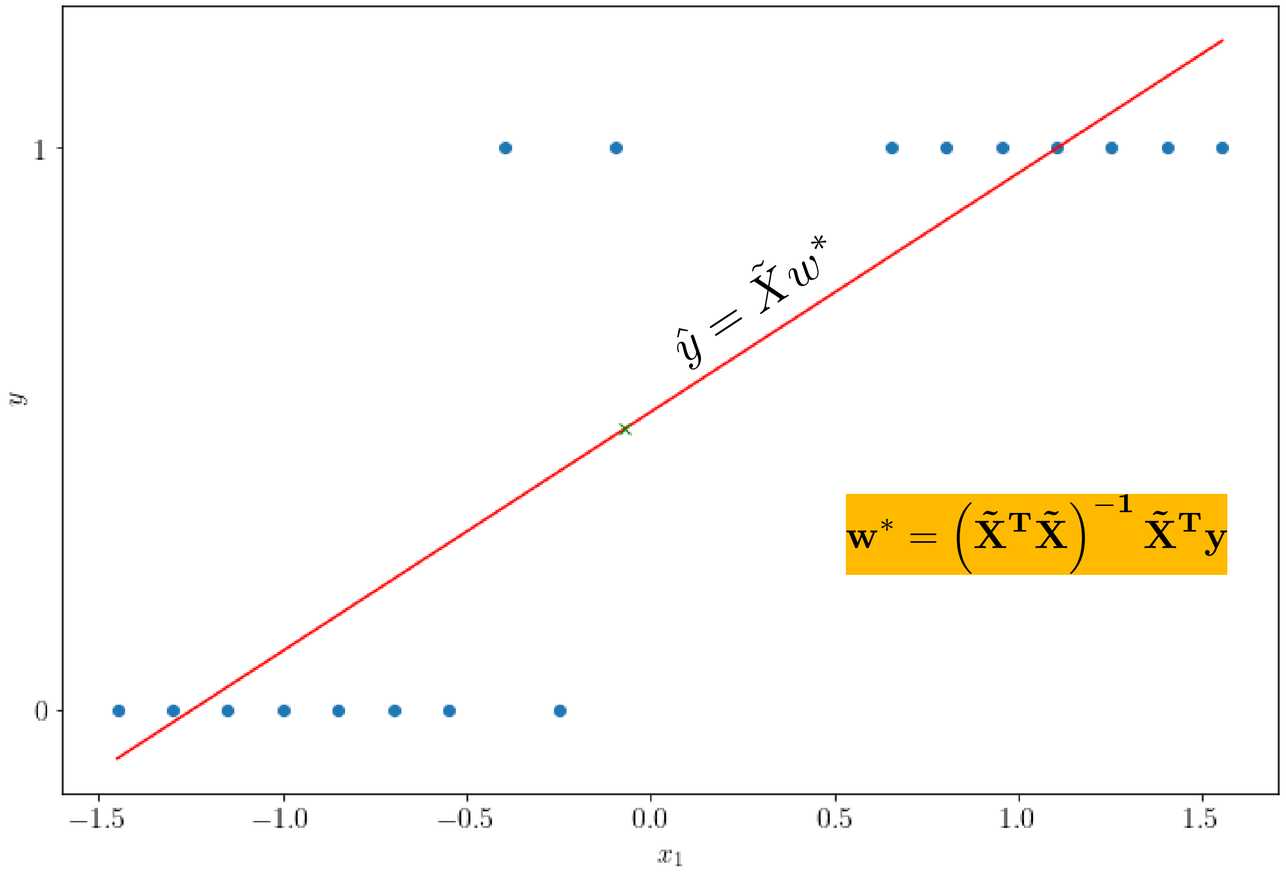
We minimize the following cost function $g(w) = \frac{1}{N} \sum_{i=1}^{N} \left(\tilde{X}^i w - y^i \right)^2$



Proposition #1

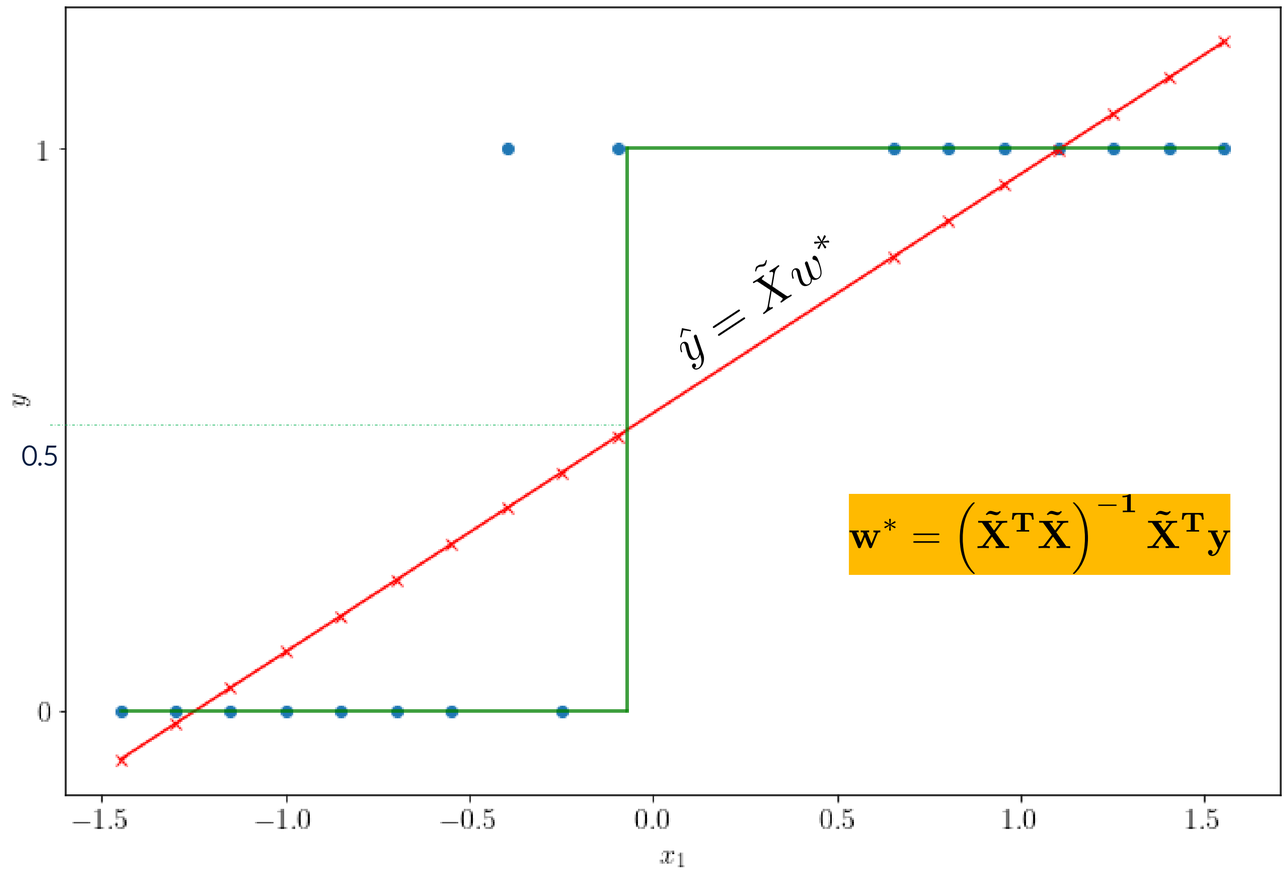
How about applying the linear regression framework seen earlier to fit such data?

We minimize the following cost function $g(w) = \frac{1}{N} \sum_{i=1}^{N} \left(\tilde{X}^i w - y^i \right)^2$



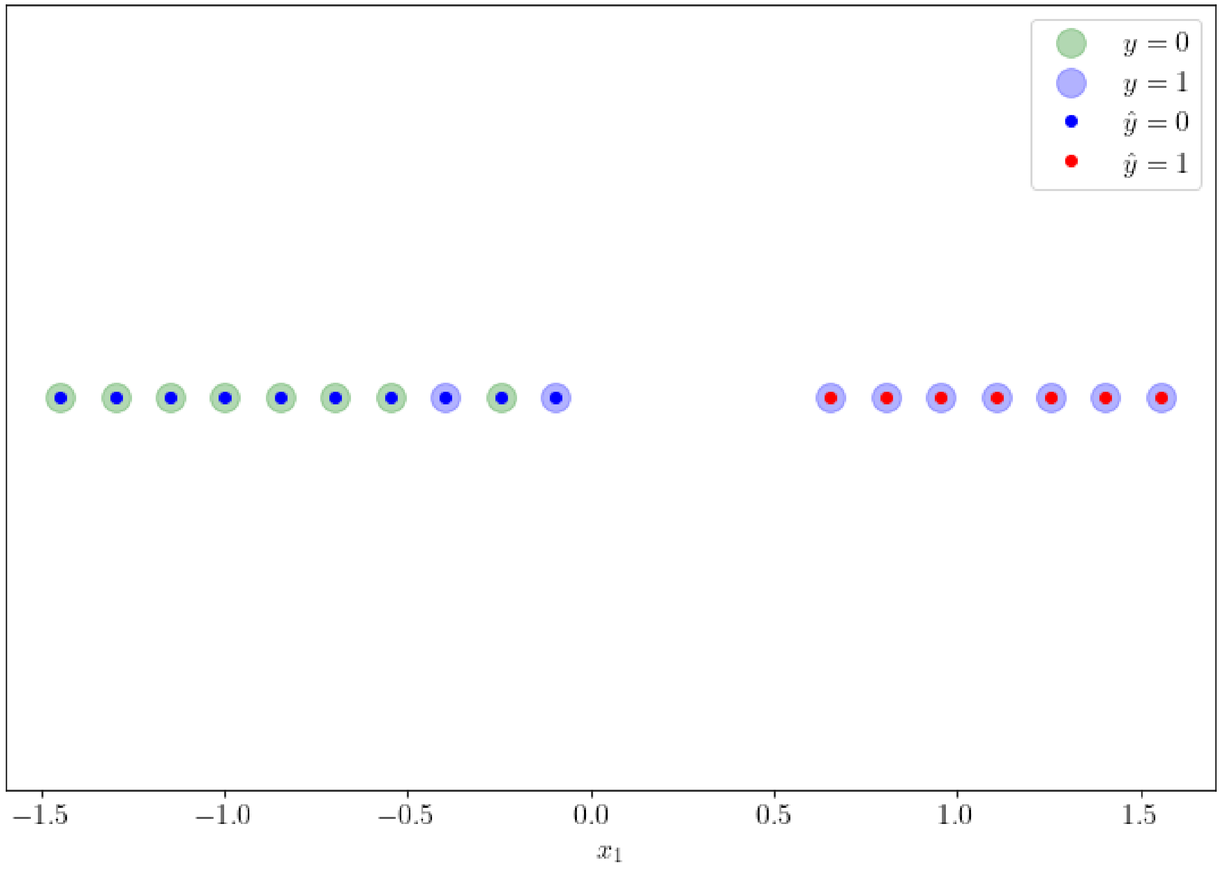
Proposition #1

And we apply a step function $\text{step}(\tilde{X}w^* - 0.5)$



Proposition #1

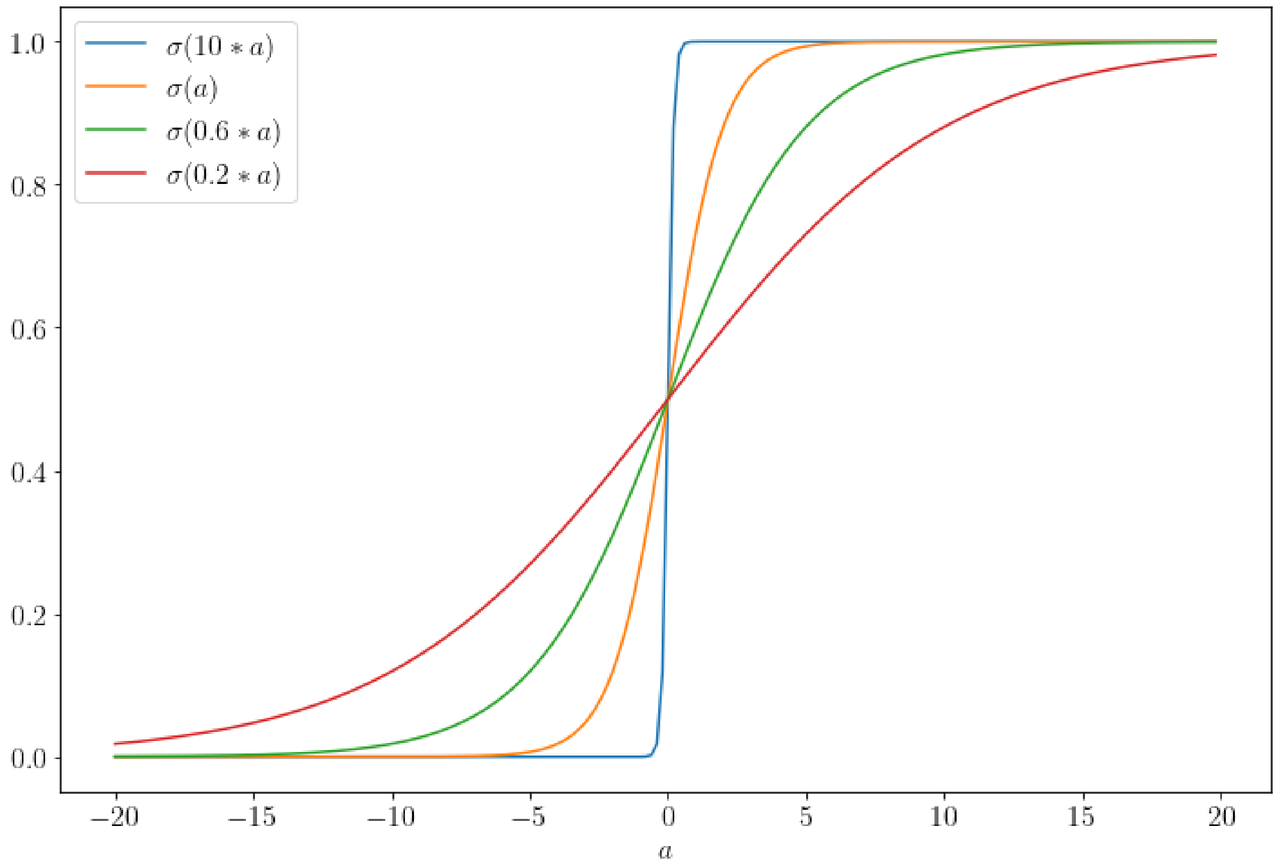
Results from the other perspective: 2 **classification errors.** In this case, 2 false negatives.



Proposition #2 – Intuition

How about integrating the step function in the cost function to minimize?

Or, better yet, a smooth approximation of the step function, the logistic sigmoïd $\sigma(x) = \frac{1}{1+e^{-x}}$



Proposition #2 – Formally

We'll explicitly model the probability $p := P(\hat{y} = 1|X)$

The probability takes values in [0, 1]. The range of a linear model is not bounded.

We'll extend the range to $[-\infty, +\infty]$, and approximate $\ln\left(\frac{p}{1-p}\right)$ by a linear model

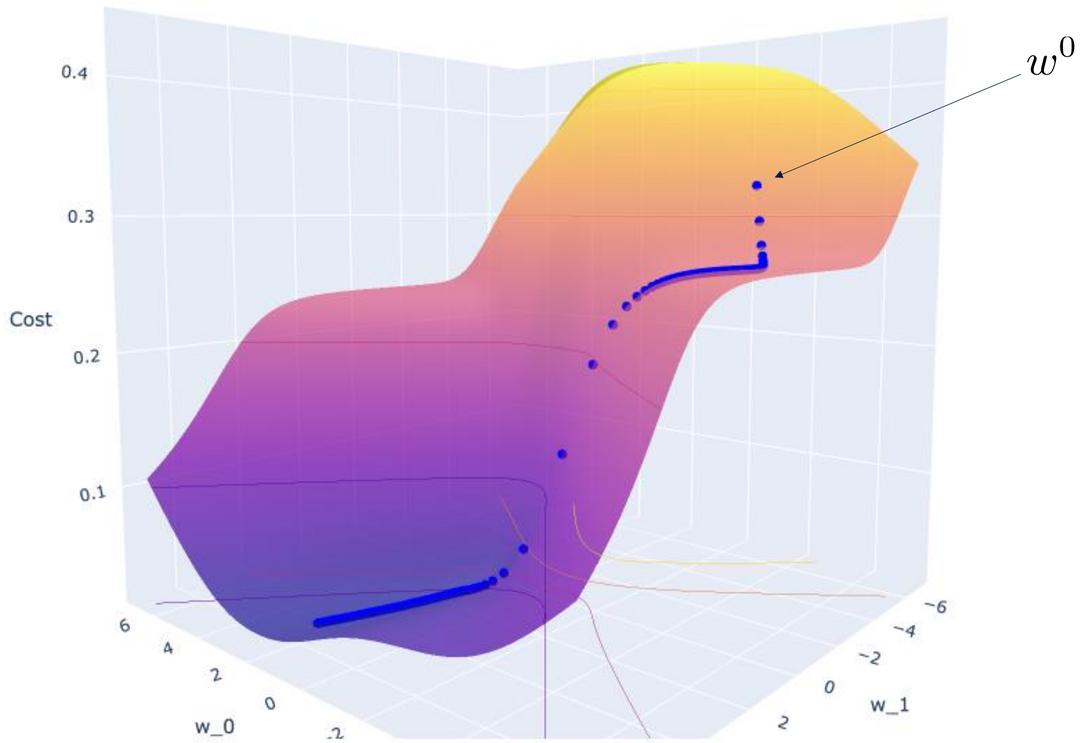
Hence, $\ln\left(\frac{p}{1-p}\right) = \tilde{X}w$, which can be written as:

$$p := P(\hat{y} = 1|X) = \frac{e^{\tilde{X}w}}{1 + e^{\tilde{X}w}} = \frac{1}{1 + e^{-\tilde{X}w}} = \sigma\left(\tilde{X}w\right)$$

Proposition #2

So we'll now minimize the following cost function $\mathcal{J}_{\text{sl}}(w) = \frac{1}{2N}\sum_{i=1}^{N}\left(\sigma\left(\tilde{X}^i w\right) - y^i\right)^2$

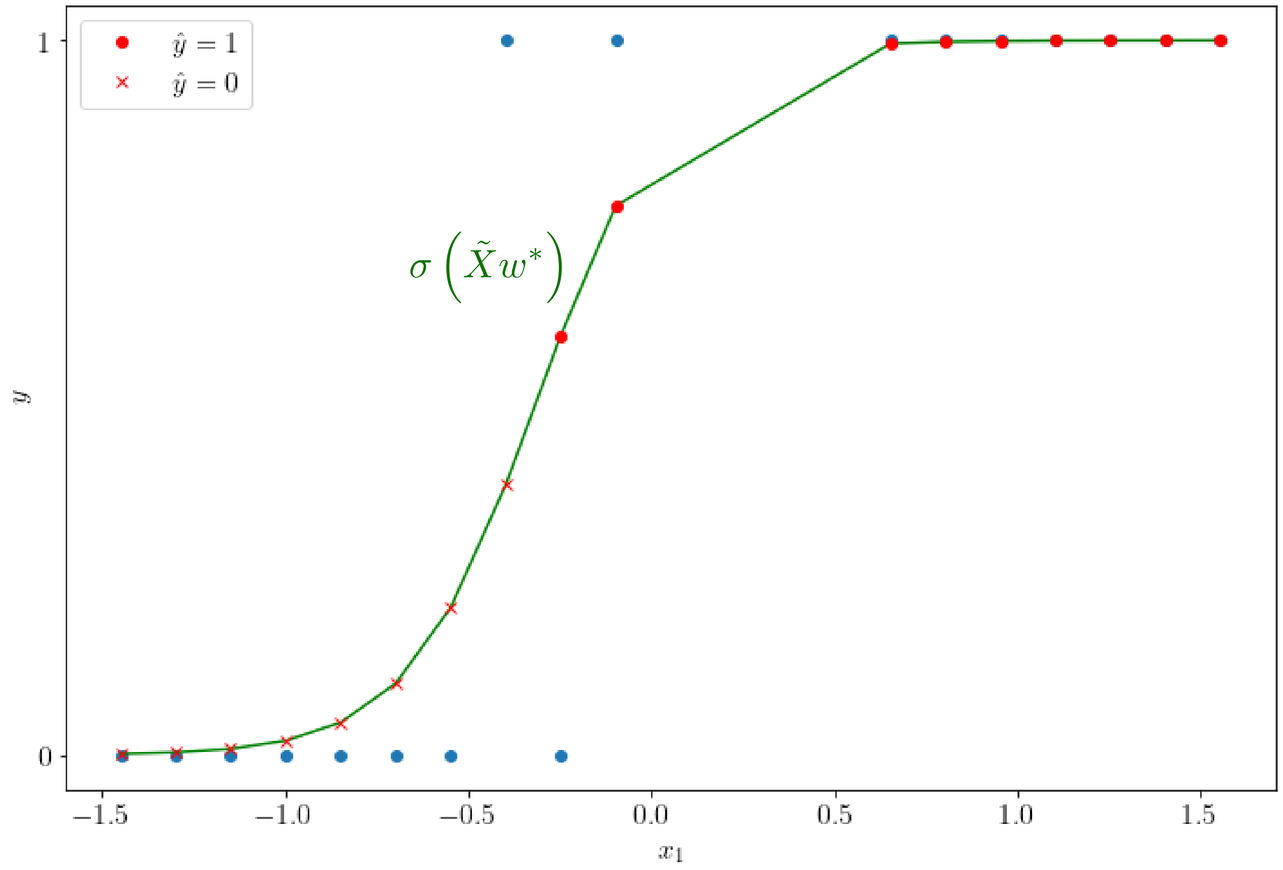
There is no closed-form solution for w^* . We'll use **gradient descent** (cf. notebook).



Proposition #2

So we'll now minimize the following cost function $\mathcal{L}_{\text{sl}}(w) = \frac{1}{2N} \sum_{i=1}^{N} \left(\sigma \left(\tilde{X}^i w \right) - y^i \right)^2$

There is no closed-form solution for w^* . We'll use **gradient descent** (cf. notebook).



Proposition #3 – Cross entropy

A random variable follows a Bernoulli distribution if it only has two possible outcomes: 0 or 1.

$$P(Y = y \mid X) = p^y \, (1 - p)^{1-y}, \quad y = \{0, \, 1\}.$$

The likelihood of a Bernouilli distribution over N observations:

$$L(p) = \prod_{i=1}^{N} p^{y^i} \, (1 - p)^{1-y^i}$$

which we wish to maximize.

Recall that $p^i = P\left(\hat{y}^i = 1 | \tilde{X}^i\right) = \sigma\left(\tilde{X}^i w\right)$

We want to minimize the negative log-likelihood:

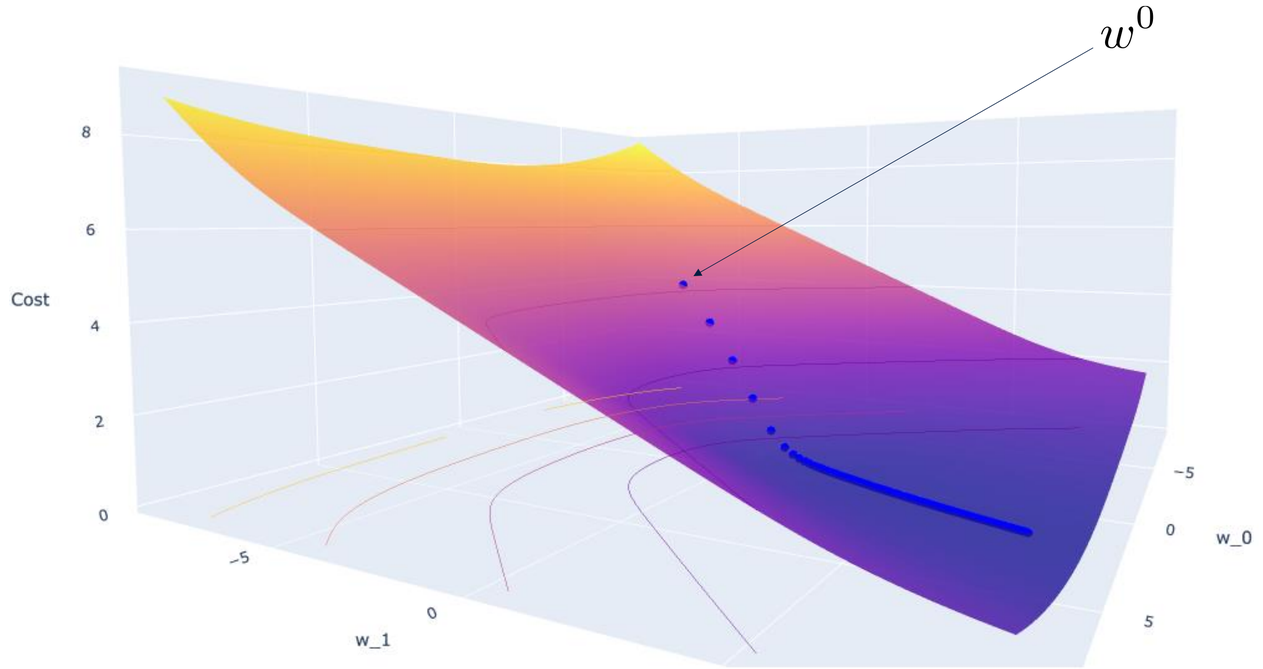
$$g_{\text{cel}} = -\frac{1}{N} \sum_{i=1}^{N} \left(y^i \ln\left(\sigma\left(\tilde{X}^i w\right)\right) + (1 - y^i) \ln\left(1 - \sigma\left(\tilde{X}^i w\right)\right)\right)$$

Proposition #3

We want to minimize the negative of the log-likelihood (or **cross entropy**):

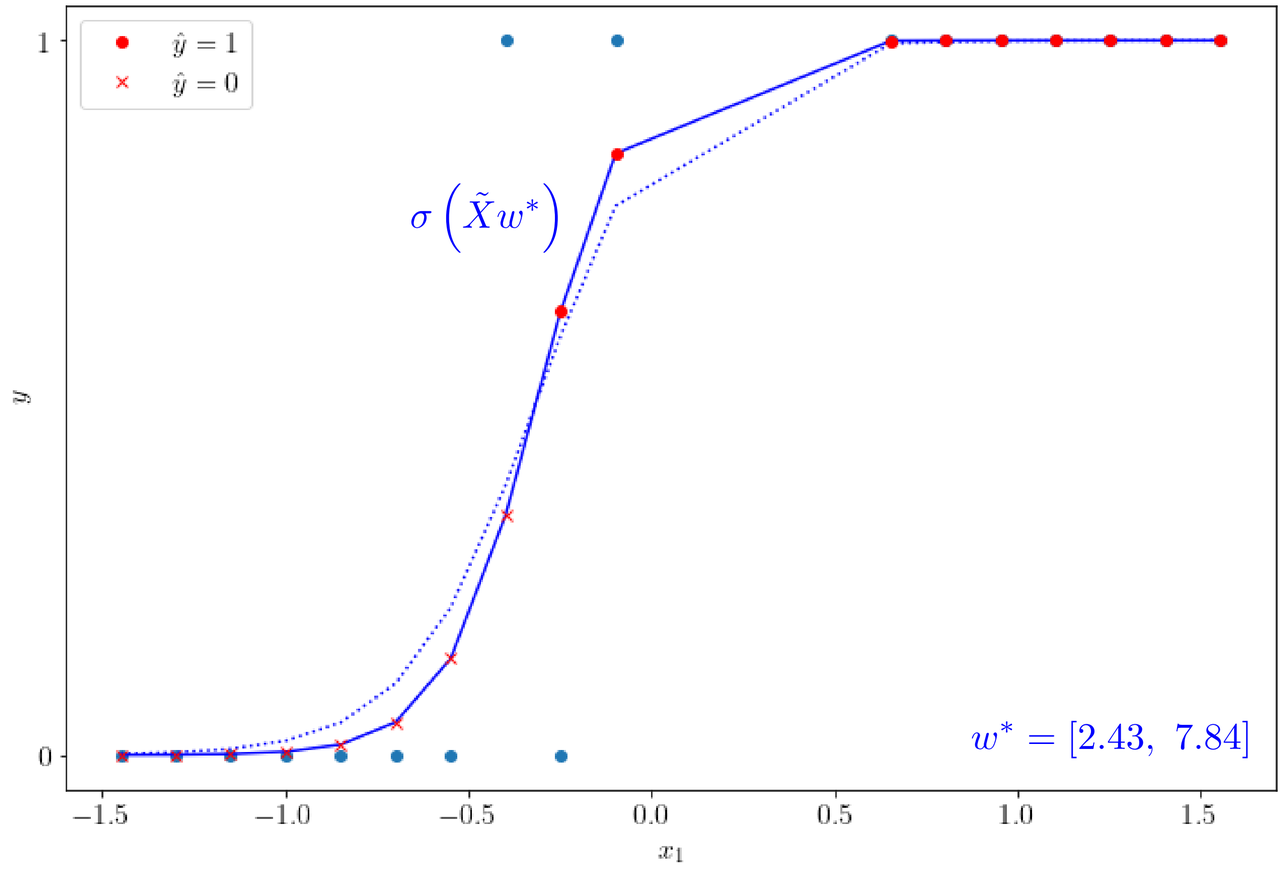
$$g_{\text{cel}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y^i \ln\left(\sigma\left(\tilde{X}^i w\right)\right) + (1-y^i)\ln\left(1-\sigma\left(\tilde{X}^i w\right)\right)\right)$$

There is no closed-form solution for w^*. We'll use **gradient descent** (cf. notebook).



Minimizing the Cross Entropy

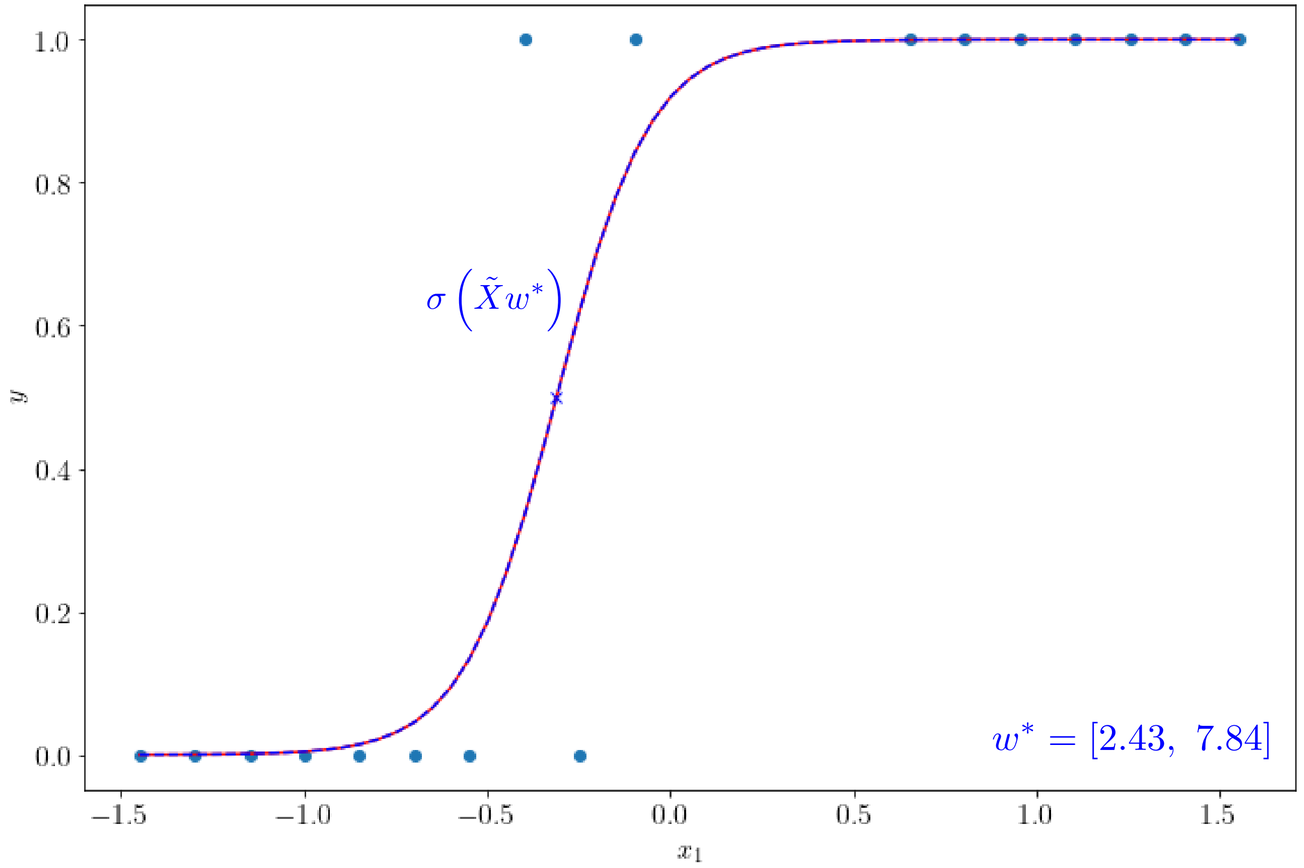
$$g_{\text{cel}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y^i \ln\left(\sigma\left(\tilde{X}^i w\right)\right) + (1-y^i)\ln\left(1-\sigma\left(\tilde{X}^i w\right)\right)\right)$$



Now using scikit-learn


```
# Régularisation: 'none' ou 'l2'
clf = sklearn.linear_model.LogisticRegression(solver='lbfgs',penalty='none')
clf.fit(X, y.reshape(N,))

w_0, w_1 = clf.intercept_[0], clf.coef_[0][0]
```



Performance evaluation

Classification

Confusion matrix and statistics

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Sensitivity: *True Positive Rate* or Recall

(1 – Specificity): *False Positive Rate*

$$\text{F1 Score} = 2 \times \frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Precision + Recall}}$$

Source: Data Science and machine learning: The confusion matrix

Exercise

- Calculate precision, sensitivity and specificity
- Calculate accuracy and F1 score

		Predicted Class: Spam	Predicted Class: Non-Spam
Actual Class	Spam	TP=45	FN=20
	Non-Spam	FP=5	TN=30

Source: Data Science and machine learning: The confusion matrix

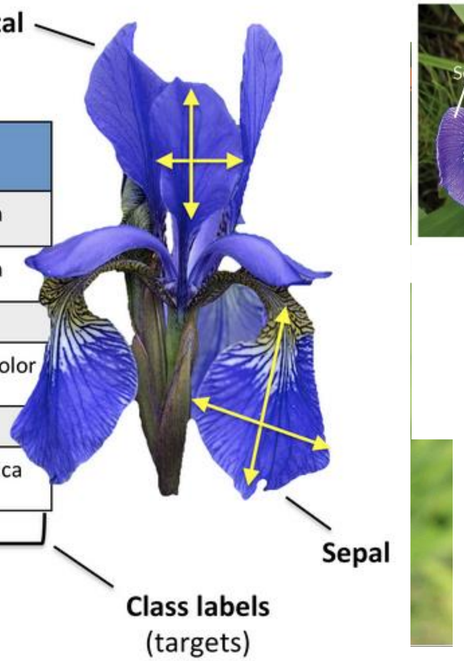
Real example using scikit-learn

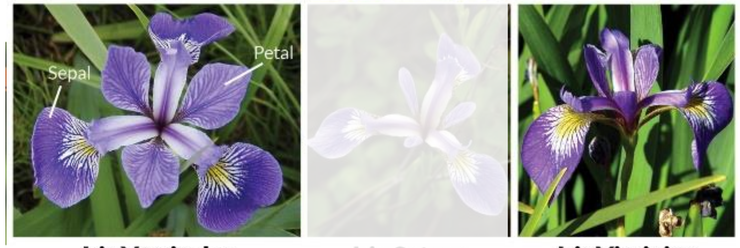
UCI Machine Learning Repository

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

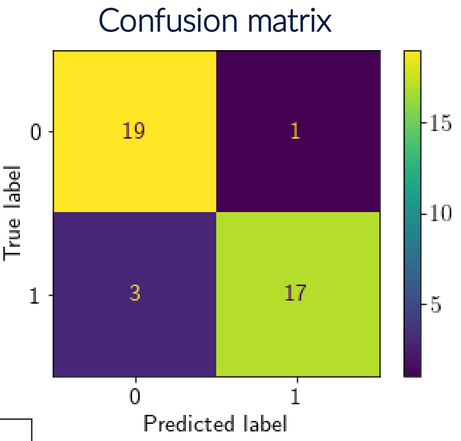


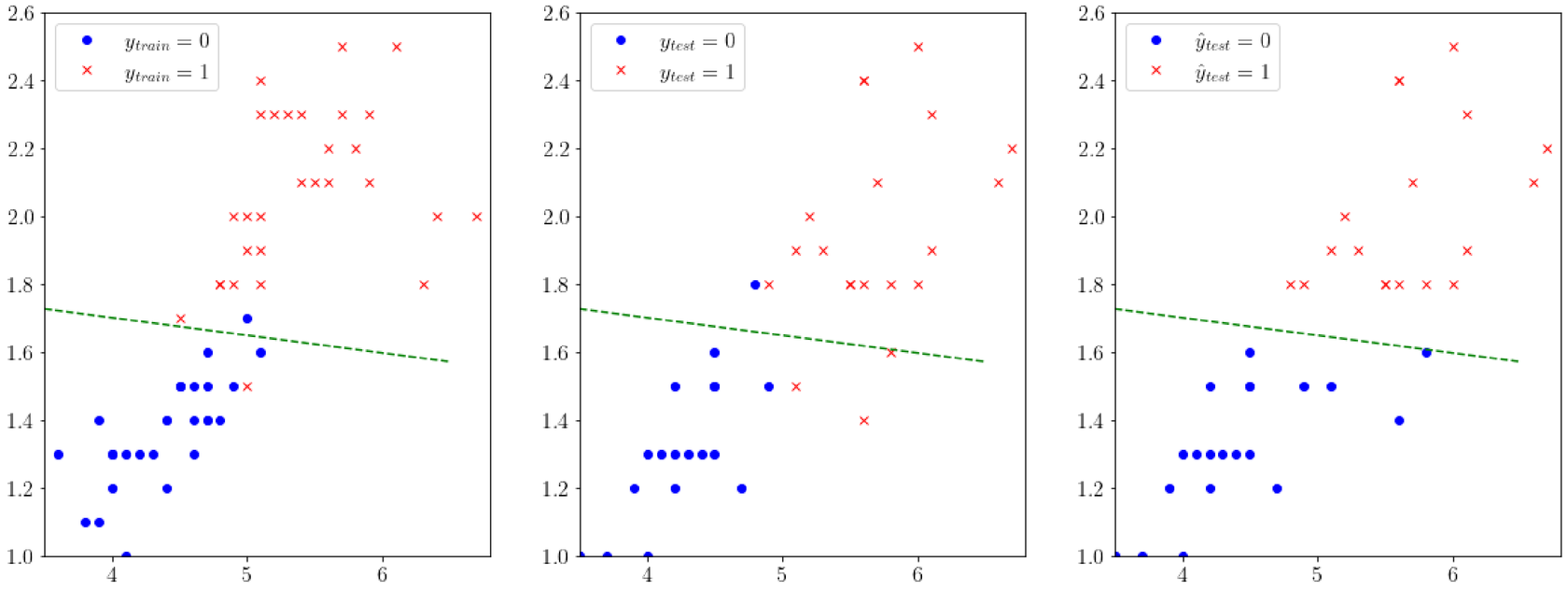


Iris Versicolor | Iris Setosa | **Iris Virginica**

Iris dataset {versicolor: 0, virginica: 1}

- Using train-test-split, 30% test
- Using logistic regression (no regularization)





What did I learn?

- Definition of (binary) classification (vs regression)
- Logistic regression using various cost functions
- In particular, the cross entropy cost (or negative log-likelihood)
- Confusion matrix for classification performance

QUESTIONS ?