Machine Learning

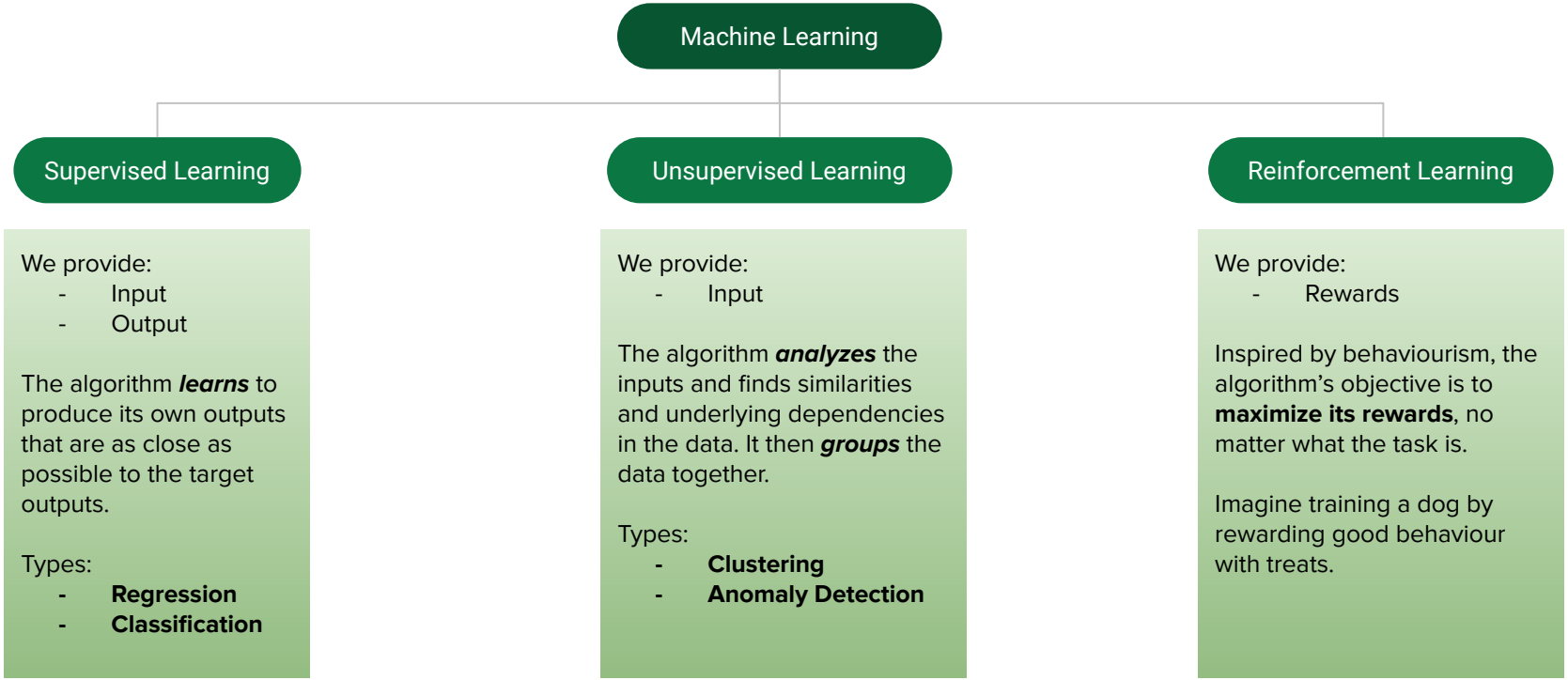
Overview

What is Machine Learning?

"A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P** improves with experience **E**."

- Tom Mitchell, *"Machine Learning"*

Types of Machine Learning



Supervised Learning Algorithms

Regression

Goal: “predict” a continuous-valued output.

E.g.: house prices, temperature, stock prices, etc.

How: Given the data described as a feature vector **x**, the model will come up with a function (named a *hypothesis*) that will map the multi-dimensional vector to a one-dimensional continuous variable, that is an estimate of the target output, y.

E.g.: $\mathbf{x} = [x_1, x_2, x_3, x_4]$

$$h_{trained} = b + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$$

where b, w_k are called *parameters* or *bias and weights*

Regression

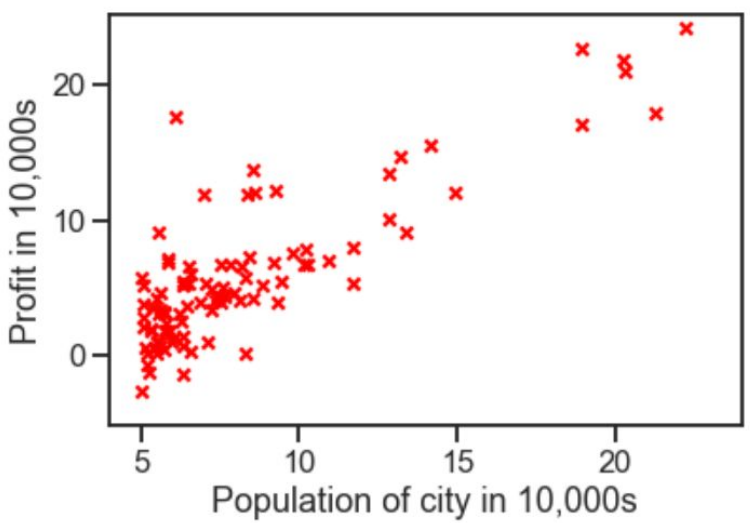
Training: Find the parameters that best **fit** the training data.

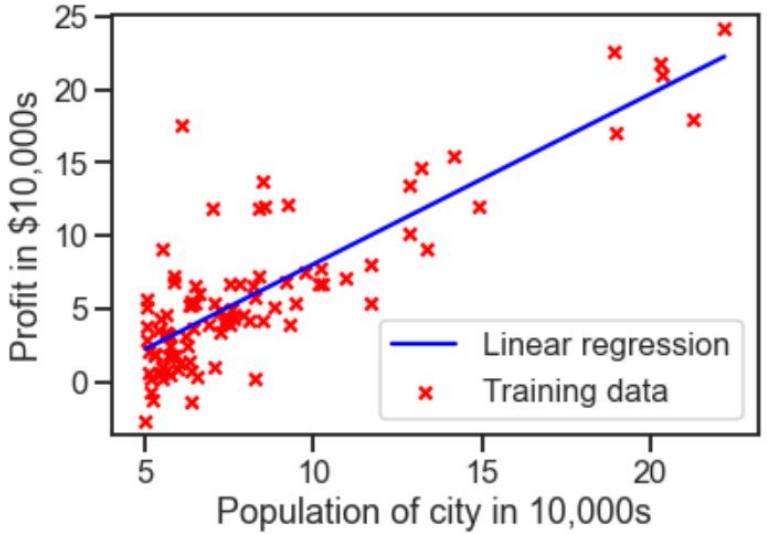
$$\text{Training Data} = \begin{bmatrix} x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, x_4^{(1)} \\ x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, x_4^{(2)} \\ .... \\ x_1^{(m)}, x_2^{(m)}, x_3^{(m)}, x_4^{(m)} \end{bmatrix} \qquad \text{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ .... \\ y^{(m)} \end{bmatrix}$$

Cost function: A measure of how wrong our hypothesis' estimates are in comparison with the target outputs.

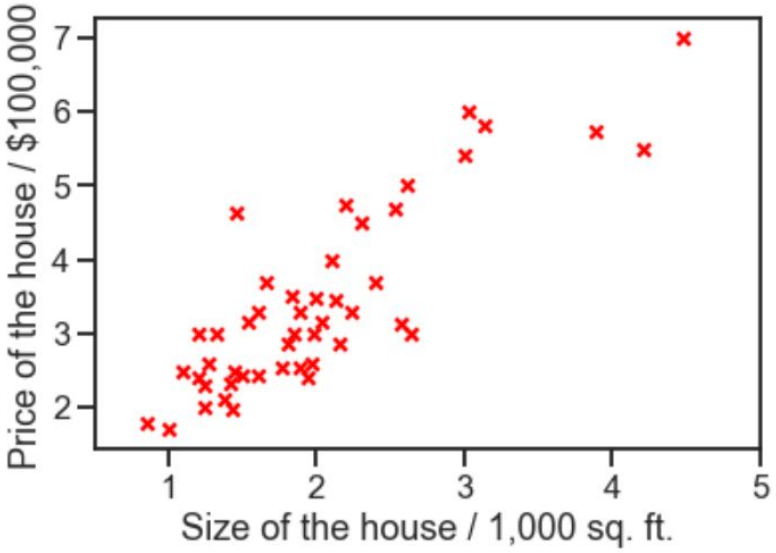
Objective: Find the bias and weights that minimize the Cost function.

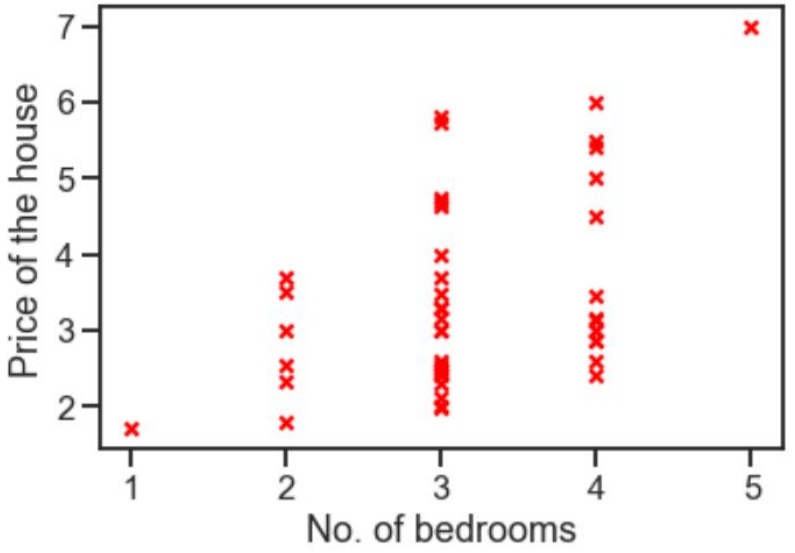
Regression


Profit in 10,000s
20
10
0
5
10
15
20
Population of city in 10,000s

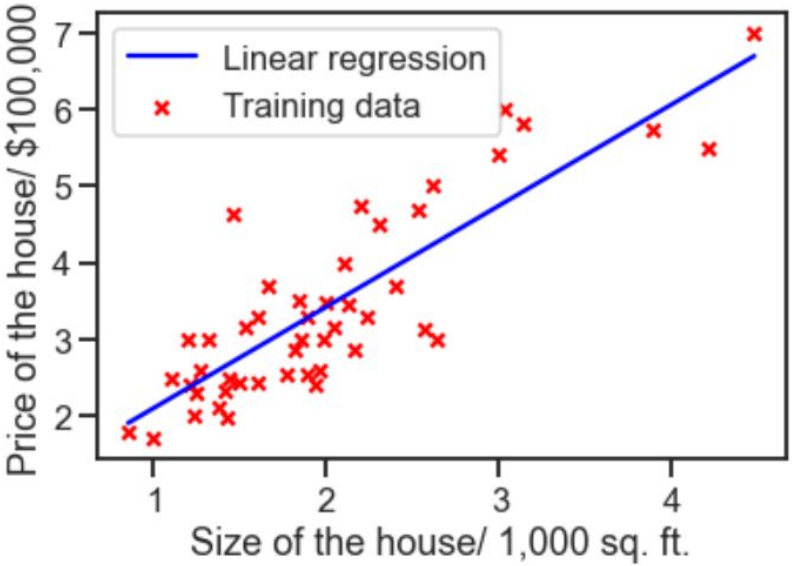

Profit in $10,000s
25
20
15
10
5
0
5
10
15
20
Population of city in 10,000s
Linear regression
Training data

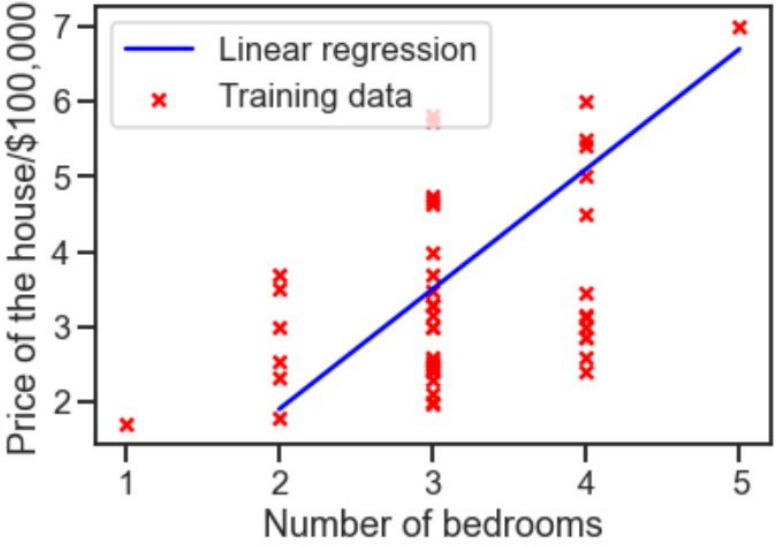
Regression


Price of the house / $100,000
7
6
5
4
3
2
1
2
3
4
5
Size of the house / 1,000 sq. ft.


Price of the house
7
6
5
4
3
2
1
2
3
4
5
No. of bedrooms

Regression


Linear regression
Training data
Price of the house/ $100,000
Size of the house/ 1,000 sq. ft.
1
2
3
4
2
3
4
5
6
7


Linear regression
Training data
Price of the house/$100,000
Number of bedrooms
1
2
3
4
5
2
3
4
5
6
7

Classification

Goal: predict a discrete-valued output.

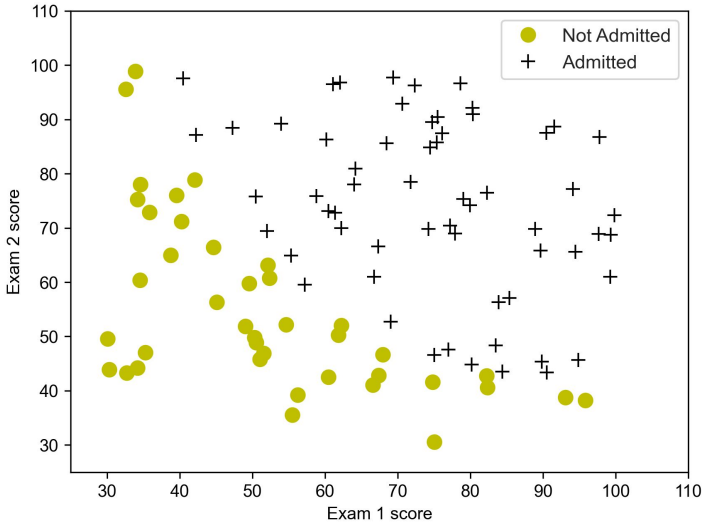
E.g.: boolean variables (pedestrian/non-pedestrian), categorical data, etc.

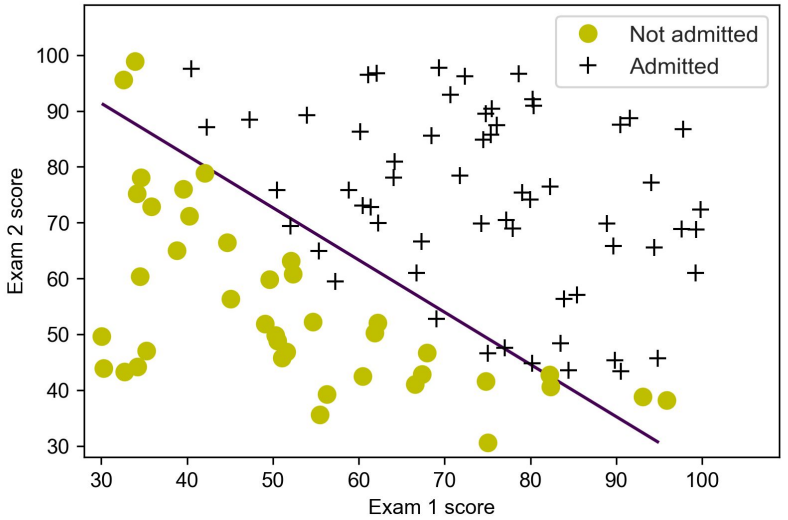
How: Mapping multi-dimensional feature vectors into a one-dimensional discrete-valued variable.

Note: The training process is similar to Linear Regression, except that the hypothesis and the Cost function take a different form.

New concept: The Decision Boundary is a line described by the hypothesis, and it separates between the two classes.

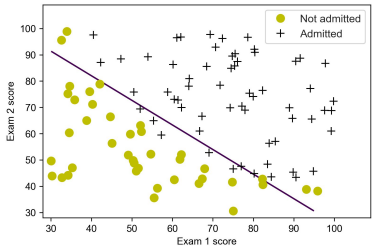
Classification





Classification

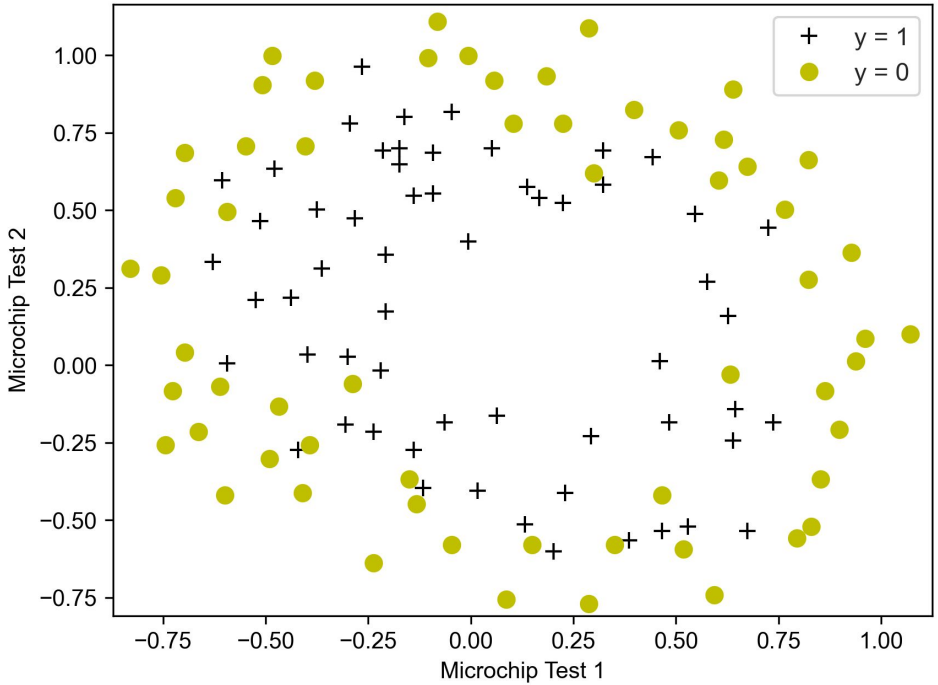
Accuracy on the training set: A measure of how many training examples the model classifies correctly.



Training set accuracy: 80.93%

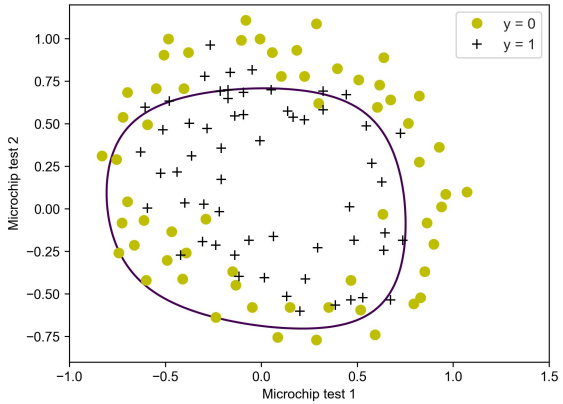
Can we increase the training set accuracy? Yes.

Classification

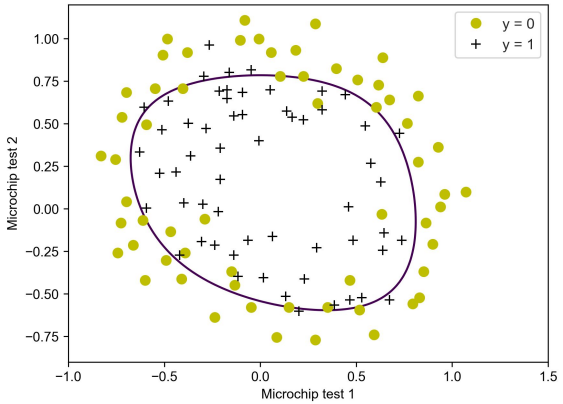


Classification

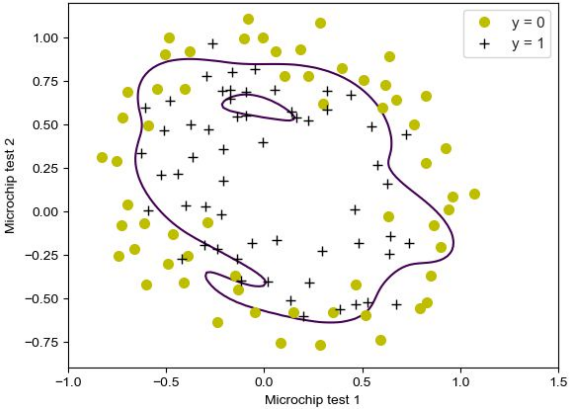
$\text{Accuracy}_{\text{train set}} = 74.6\%$



$\text{Accuracy}_{\text{train set}} = 83.1\%$

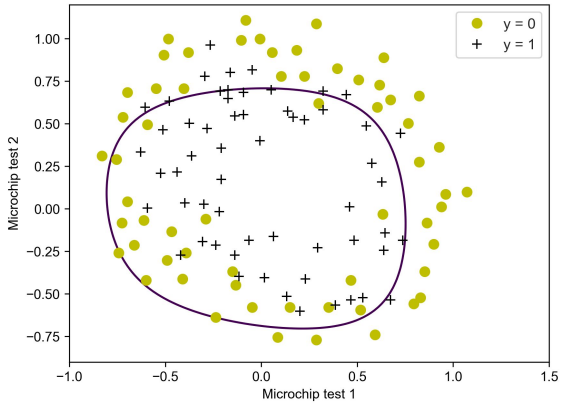


$\text{Accuracy}_{\text{train set}} = 88.1\%$



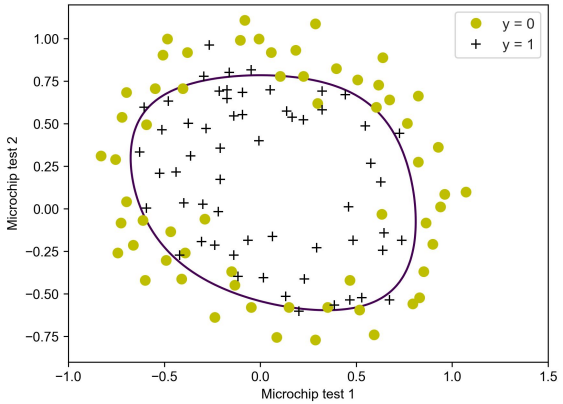
Classification

$\text{Accuracy}_{\text{train set}} = 74.6\%$

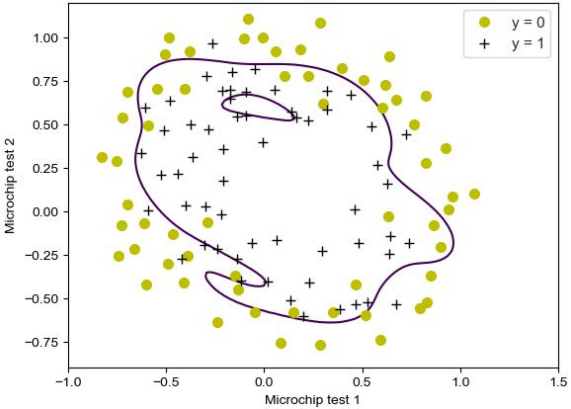


Underfit (High bias)

$\text{Accuracy}_{\text{train set}} = 83.1\%$

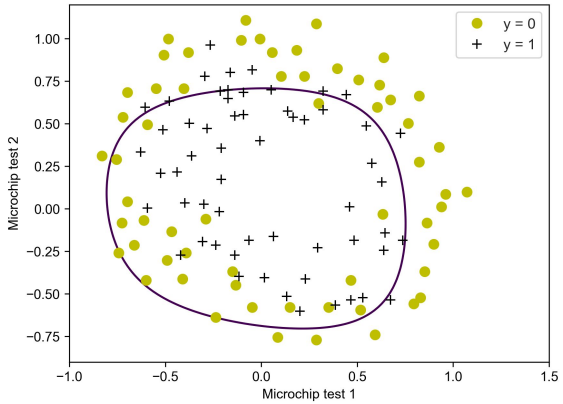


$\text{Accuracy}_{\text{train set}} = 88.1\%$



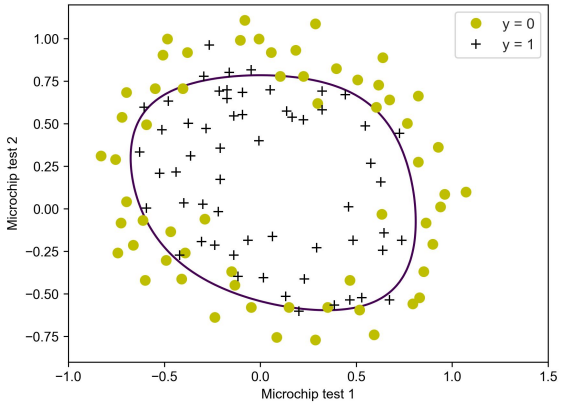
Classification

$\text{Accuracy}_{\text{train set}} = 74.6\%$

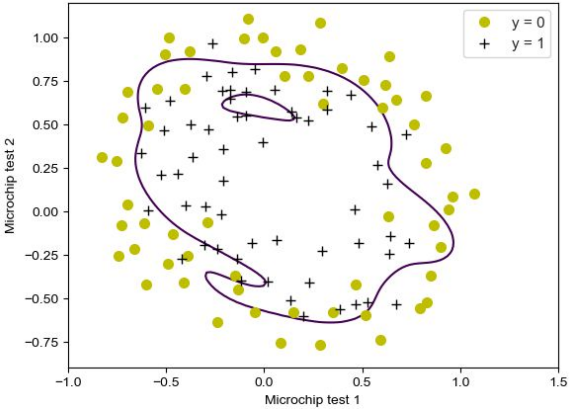


Underfit (High bias)

$\text{Accuracy}_{\text{train set}} = 83.1\%$



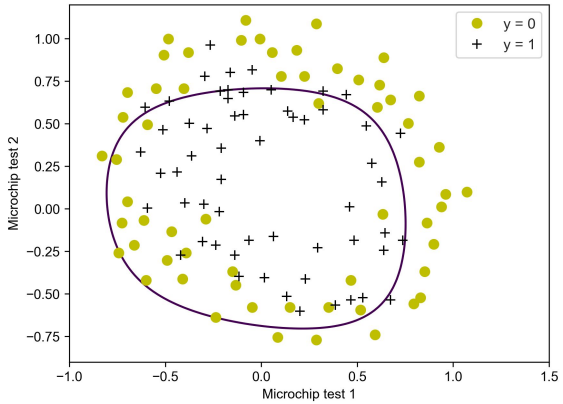
$\text{Accuracy}_{\text{train set}} = 88.1\%$



Overfit (High variance)

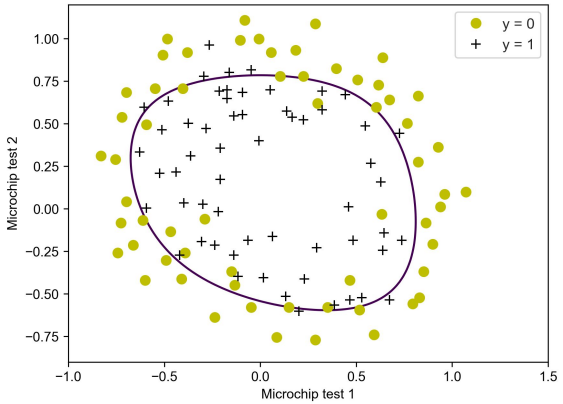
Classification

$\text{Accuracy}_{\text{train set}} = 74.6\%$



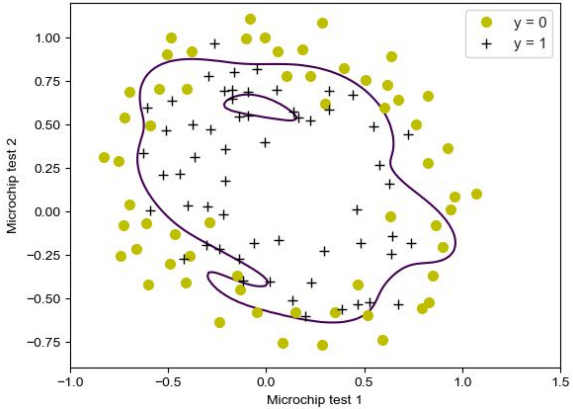
Underfit (High bias)

$\text{Accuracy}_{\text{train set}} = 83.1\%$



Just right
(Bias/Variance Trade-off)

$\text{Accuracy}_{\text{train set}} = 88.1\%$



Overfit (High variance)

Bias/Variance Trade-off

Training Set Accuracy is a poor performance indicator for classifiers.

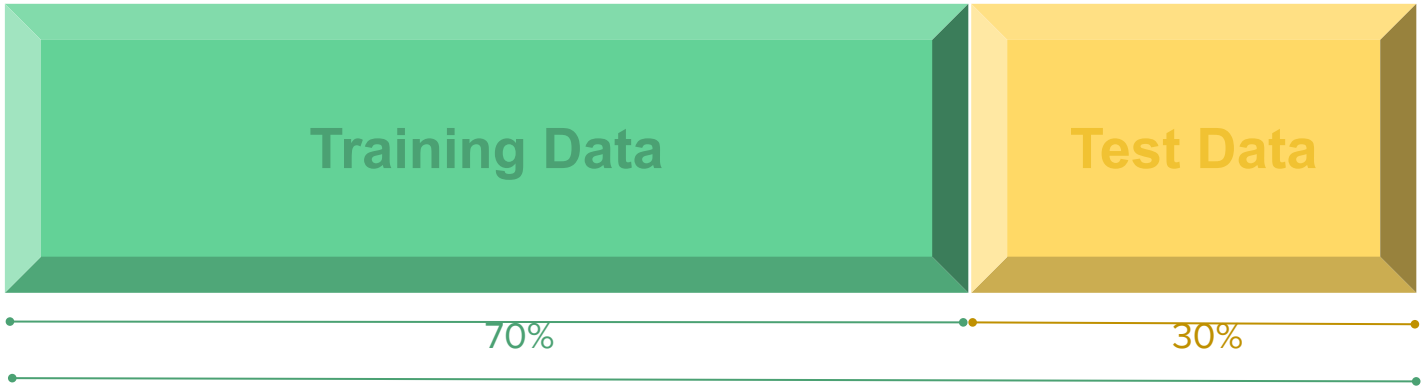
Accuracy on test set vs. Accuracy on training set: a good indicator of Overfitting/Underfitting problems.

Bias/Variance Trade-off

Training Set Accuracy is a poor performance indicator for classifiers.

Accuracy on test set vs. Accuracy on training set: a good indicator of Overfitting/Underfitting problems.

Data:

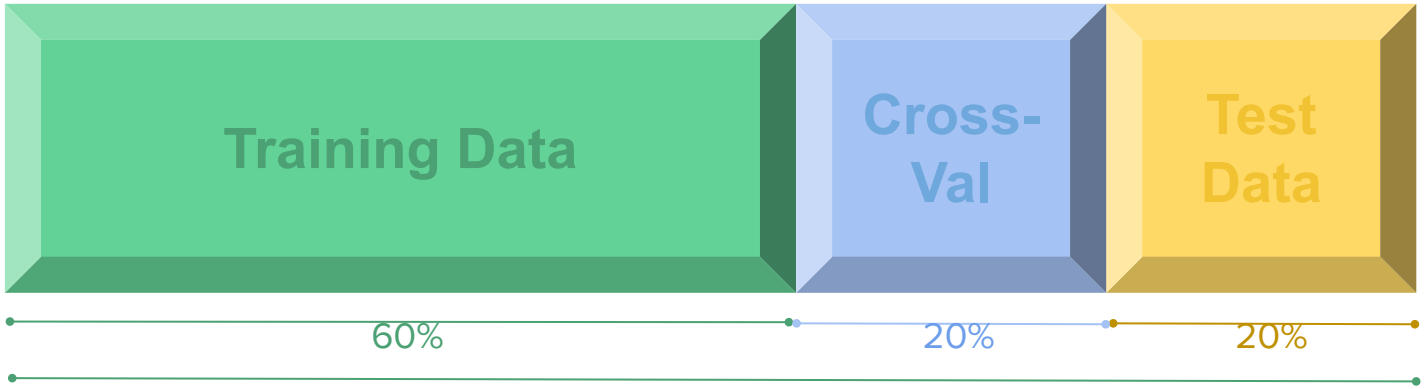


Bias/Variance Trade-off

Training Set Accuracy is a poor performance indicator for classifiers.

Accuracy on test set vs. Accuracy on training set: a good indicator of Overfitting/Underfitting problems.

Data:



Garbage-in, garbage-out

"A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P** improves with experience **E**."

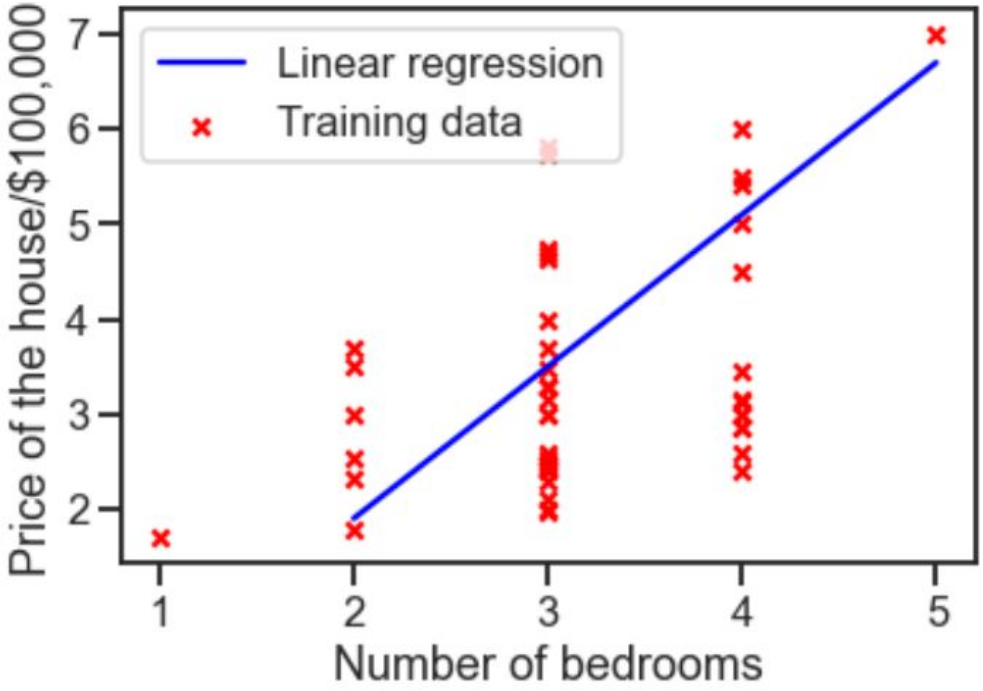
- Tom Mitchell, *"Machine Learning"*

But...

"If the experience **E** is unrelated to the task **T**, then the performance measure **P** will be an inaccurate representation of the machine's learning."

- Stefan Mihai, and I'm sure many others

Garbage-in, garbage-out



Meeting Insights

- The ML model is only as good as the data we train/validate/test it with
- The acquired data set must be large enough to allow splitting into a training set, a validation set, and a test set
- The Bias/Variance Trade-Off is continuously present in all ML models

Thank you!