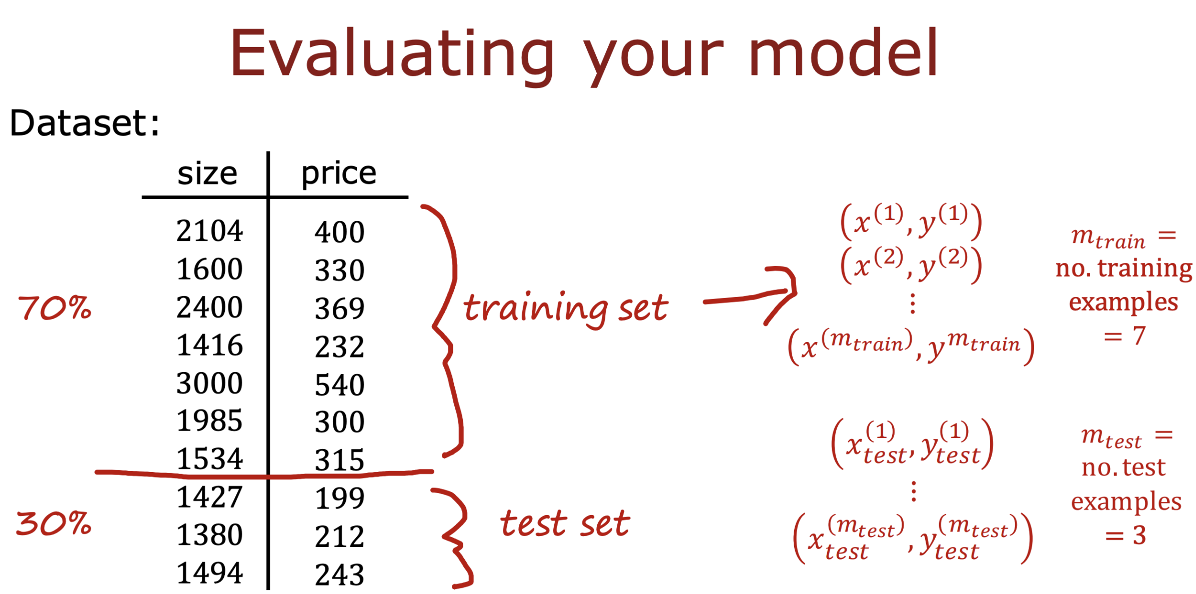
Advice for applying machine learning

Diagnostic: A test that you run to gain insight into what is/isn't working with a learning algorithm, to gain guidance into improving its performance.

Evaluating model

Train/test sets

Splitting the training set into two subsets: a training set (70% of the data) and a test set (30% of the data). The model parameters are trained on the training set, and its performance is tested on the test set.



Train/test procedure for linear regression (with squared error cost)

Fit parameters by minimizing cost function $J(\vec{w}, b)$

$$J(\vec{w},b) = \min_{\vec{w},b}\left[\frac{1}{2m_{train}}\sum_{i=1}^{m_{train}}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2m_{train}}\sum_{j=1}^{n} w_j^2\right]$$

Compute test error:

$$J_{test}(\vec{w},b) = \frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}_{test}\right) - y^{(i)}_{test}\right)^2\right] \quad \cancel{+\sum w^2}$$

Compute training error:

$$J_{train}(\vec{w},b) = \frac{1}{2m_{train}}\left[\sum_{i=1}^{m_{train}}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}_{train}\right) - y^{(i)}_{train}\right)^2\right]$$

Train/test procedure for classification problem

0/1

Fit parameters by minimizing $J(\vec{w},b)$ to find $\vec{w}, b$

E.g.,

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\right.\right.$$

Compute test error:

$$J_{test}(\vec{w},b) = -\frac{1}{m_{test}}\sum_{i=1}^{m_{test}}\left[y\right.$$

Compute train error:

$$J_{train}(\vec{w},b) = -\frac{1}{m_{train}}\sum_{i=1}^{m_{train}}$$

fraction of the test set and the fraction of the train set that the algorithm has misclassified.

$$\hat{y} = \begin{cases} 1 \text{ if } f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \geq 0.5 \\ 0 \text{ if } f_{\vec{w},b}\left(\vec{x}^{(i)}\right) < 0.5 \end{cases}$$

count $\hat{y} \neq y$

$J_{test}(\vec{w},b)$ is the fraction of the test set that has been misclassified.

$J_{train}(\vec{w},b)$ is the fraction of the train set that has been misclassified.

fitting the parameters by minimising the cost function and then computing the test error and training error using the logistic loss on the test and training data. But an alternative and easy method of computing the test and training error by measuring the fraction of the test and training set that the algorithm has misclassified.

Once the parameter are fit using training data, then the training error will likely to be much lower than generalisation error.

The test error is better estimate of how well the model will generalise to new data.

Model Selection (choosing a model)

We can further refine this evaluating model by introducing the concept of a cross-validation set, which is used to check the accuracy of different models.

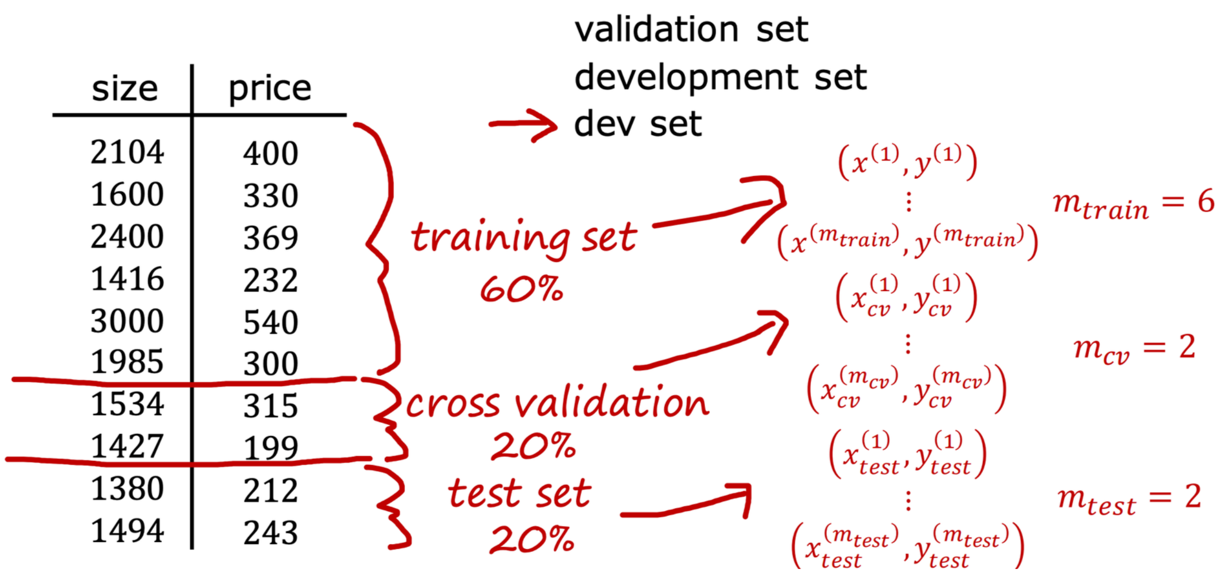
The data is split into three subsets:

the training set : to fit every model

the cross-validation set : to choose the best performing model

the test set : to get estimate of how model will perform on unknown data (generalisation error)

The model with the lowest cross-validation error is chosen.



Training/cross validation/test set

Training error: $$J_{train}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{train}}\left[\sum_{i=1}^{m_{train}}\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}^{(i)}\right) - y^{(i)}\right)^2\right]$$

Cross validation error: $$J_{cv}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{cv}}\left[\sum_{i=1}^{m_{cv}}\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{cv}^{(i)}\right) - y_{cv}^{(i)}\right)^2\right]$$ (validation error, dev error)

Test error: $$J_{test}(\vec{\mathrm{w}}, b) = \frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}}\left(f_{\vec{\mathrm{w}},b}\left(\vec{\mathrm{x}}_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2\right]$$