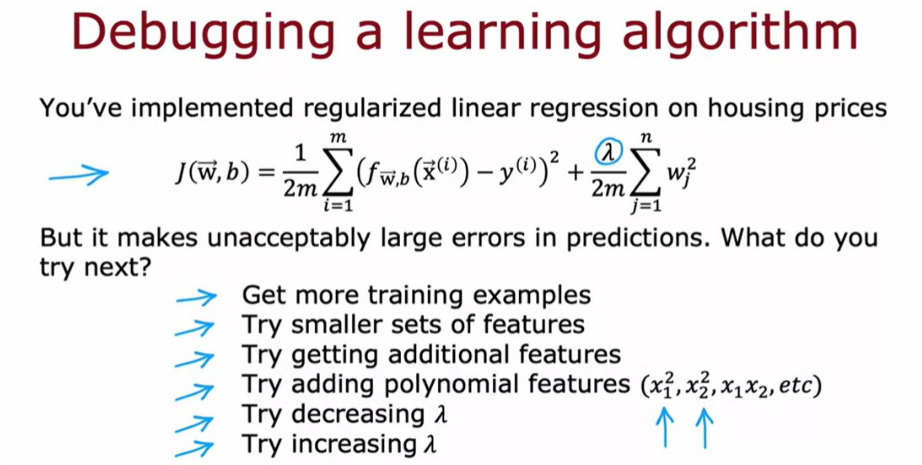
Week 3

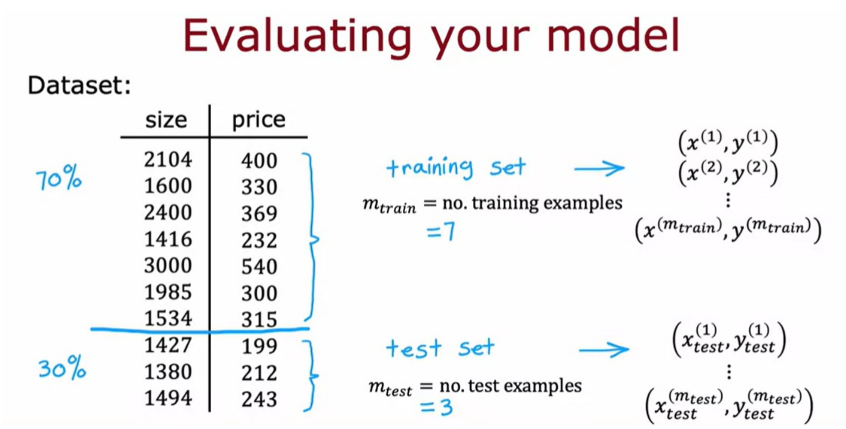
Advice for Machine Learning

Debugging



Evaluating a model

- Splitting datasets into train set and test set (Prevent overfitting etc.)



Computing Test and Train set error

- For Linear Regression:
- Notice that the regularization term is removed

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

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

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

Compute test error:

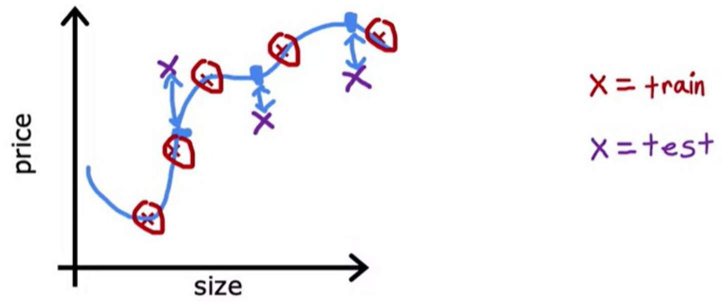
$$J_{test}(\vec{w},b)=\frac{1}{2m_{test}}\left[\sum_{i=1}^{m_{test}}\left(f_{\vec{w},b}\left(\vec{x}_{test}^{(i)}\right)-y_{test}^{(i)}\right)^2\right] \quad \cancel{\sum_{j=1}^{n}w_j^2}$$

Compute training error:

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

- Check cost of test set to see whether model is effective

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



$J_{train}(\vec{w},b)$ will be low

$J_{test}(\vec{w},b)$ will be high

- For Classification problems:
  - You can choose to compute the loss, or compute the fraction of the train/test set that was misclassified

Train/test procedure for classification problem

0/1

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

E.g.,

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

Compute test error:

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

Compute train error:

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

Train/test procedure for classification problem

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

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

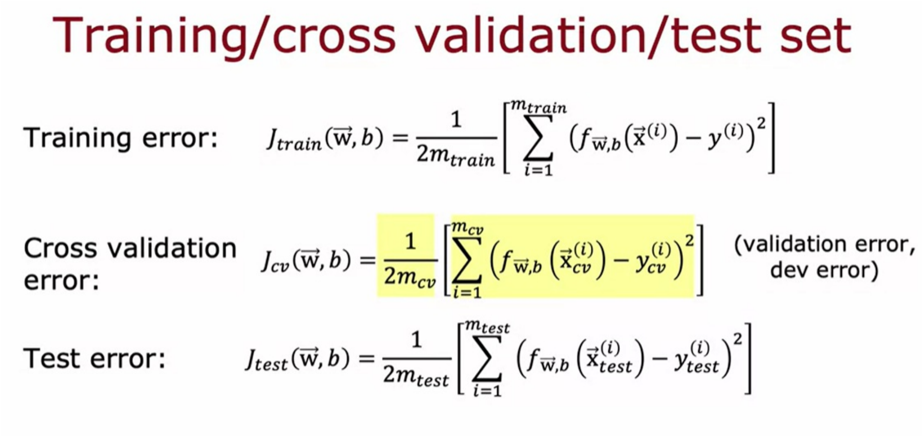
count $\hat{y}\neq y$

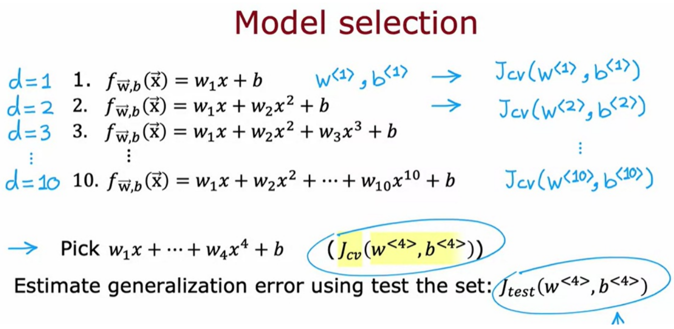
$J_{test}(\vec{w},b)$ is the fraction of the test set that has been misclassified.

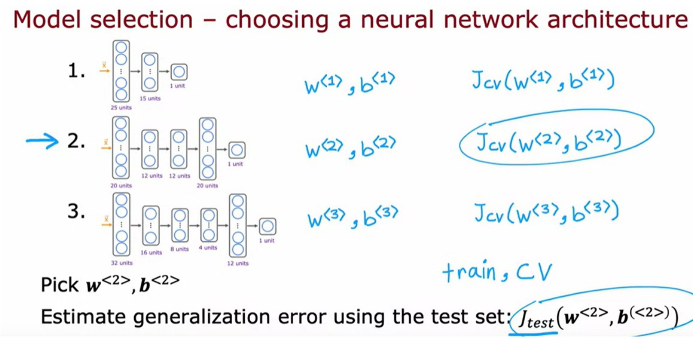
$J_{train}(\vec{w},b)$ is the fraction of the train set that has been misclassified.

Cross-validation set

- Helps with model selection and fine-tuning parameters
- Pick the model with the lowest cross validation error





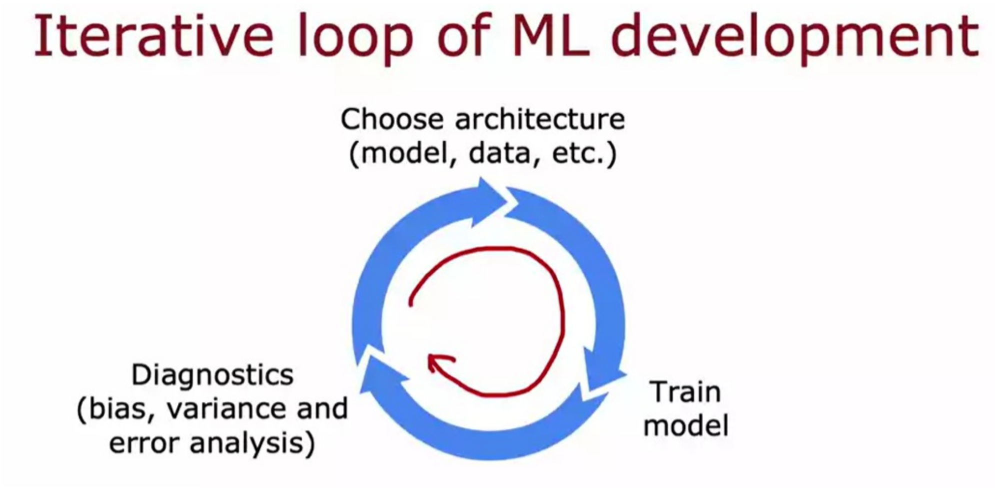


Bias and Variance

{TBC}

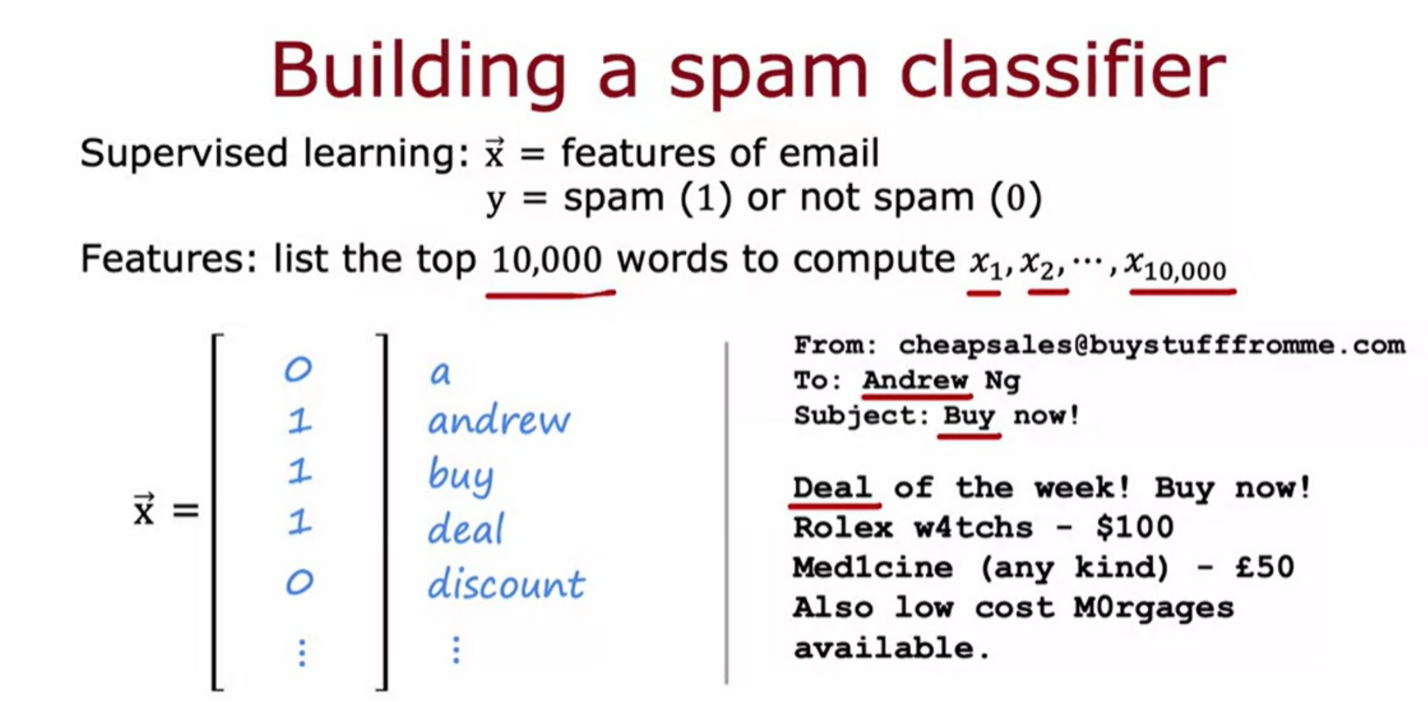
ML Development Process

Iterative Loop



Example with Spam Classification

- You can also manually spot the errors (Printing the values etc) checking the words where the model fails and misclassifies



Adding Data