Machine Learning Foundations – University of Washington

# Install Tools

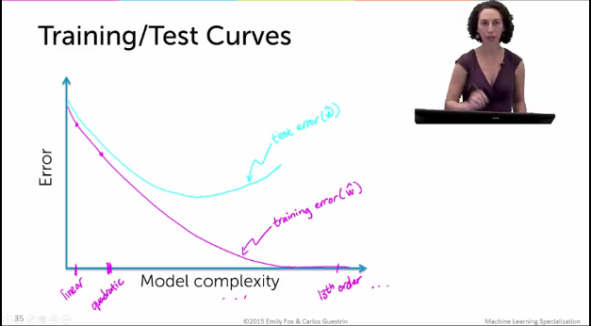
* Install GraphLab Create
  + It is a highly scalable machine learning library for Python
  + <https://dato.com/learn/coursera/>
  + installs all dependencies in an isolated environment by default
  + Installs the Anaconda python distribution
    - which includes over 300 packages for science, math, engineering, and data analysis
  + Installs iPython Notebook
    - A combination of python and a wiki page
  + SFrame package is included
    - Preferred over Pandas, because it can only hold datasets in memory
    - SFrame is out-of-core or designed to process data that is too large to fit into a computer’s main memory at one time.

# Linear Regression Modeling

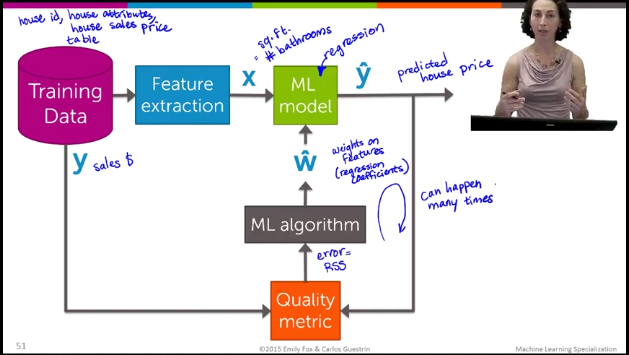
## Model Description

* Best line minimizes residual sum of squares
* “F parameterized by weight estimates”
* Adding quadratic terms, but this is still called linear regression
* A 13th order polynomial can eliminate any error, but that is not a good model

## Evaluating overfitting via training/test split

* Test set: Remove some data samples and test using those samples
* Training set: Set model parameters using the training set
* Training / Test Curves
  + As model order increases training error decreases
  + Test error decreases for lower order polynomials and then increases after the model starts to overfit
  + 

## Regression Pipeline

* 

# Classification

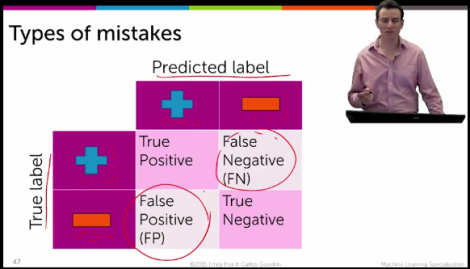
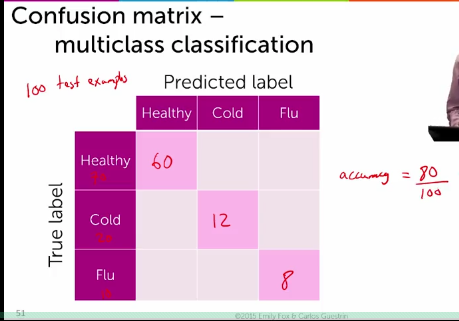
## Examples

* Sentiment Analysis: positive or negative reviews
* Spam vs. Not Spam

## Linear Classifier (regarding sentiment analysis)

* Simple: Define words as positive or negative, then compare sums
* Assign a score to each word and fit a linear decision boundary on our plot of positive vs. negative

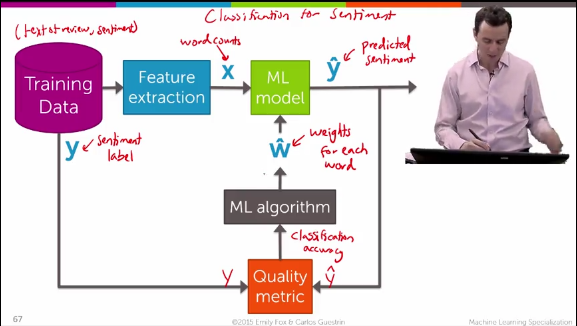
## Error and Accuracy

* Error = # wrong / # of sentences
* Accuracy = # of correct / # of sentences
* What if the sample set classes are imbalanced?
  + 90% of emails are spam, so always predicting spam would have 90% accuracy
  + 
  + 

## Learning Curves

* The more data the better if it is clean
* Plot test error vs. amount of training data
* More complex models will have higher training error at first, but lower later on.

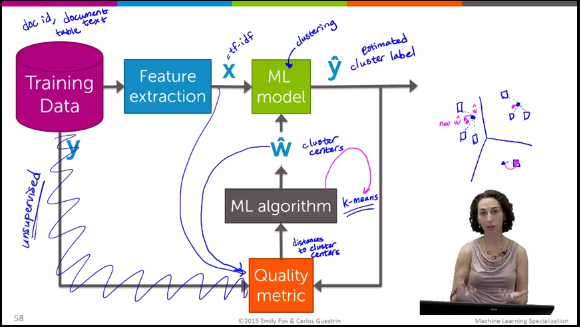
## Classification Pipeline



# Clustering

* Clustering documents
  + Divide word vector by norm of document
  + Multiply to norm’ed vectors element wise and sum to find similarity
  + TF-IDF
    - Term Frequency - Inverse Document Frequency
    - scale down common words and scale up rare words

## Clustering Pipeline

* 
* Note there is no y vector

# Recommender Systems

* Recommending a subset of products to your users
* Amazon, Netflix, Youtube
* Browsing is “history” need new ways to discover content