Before Deep Learning

Let’s talk about Machine Learning first

Slides fork of kastnerkyle.

What to expect in this workshop

https://ideascongress.files.wordpress.com/2016/03/preview-copy.jpg?w=772

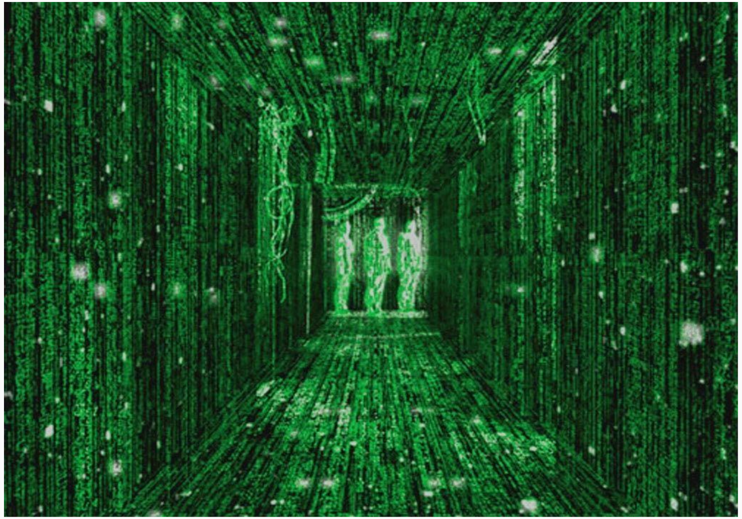
- Give you a “flavor” of what Machine Learning is about
- Only surface level concepts (not a lot of math)
- Hands-on practice of “supervised” algorithms
- A few practical tips

Why use ML?

- Drowning in data.
- Computers are cheap (and less emotional), humans are expensive.
- Psychic superpowers (sometimes)

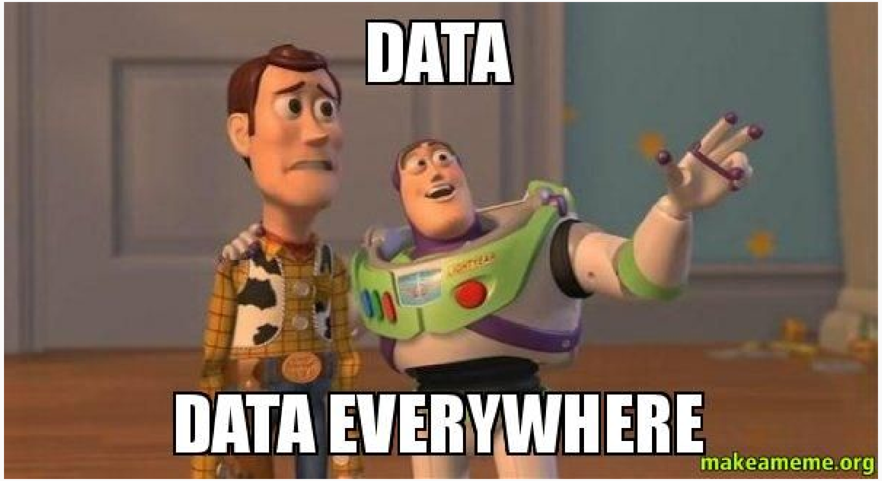
Why use ML?

- Regression (Supervised)
  - Predict housing prices
- Classification (Supervised)
  - Handwritten digit recognition
- Clustering (Unsupervised)
  - Document tagging

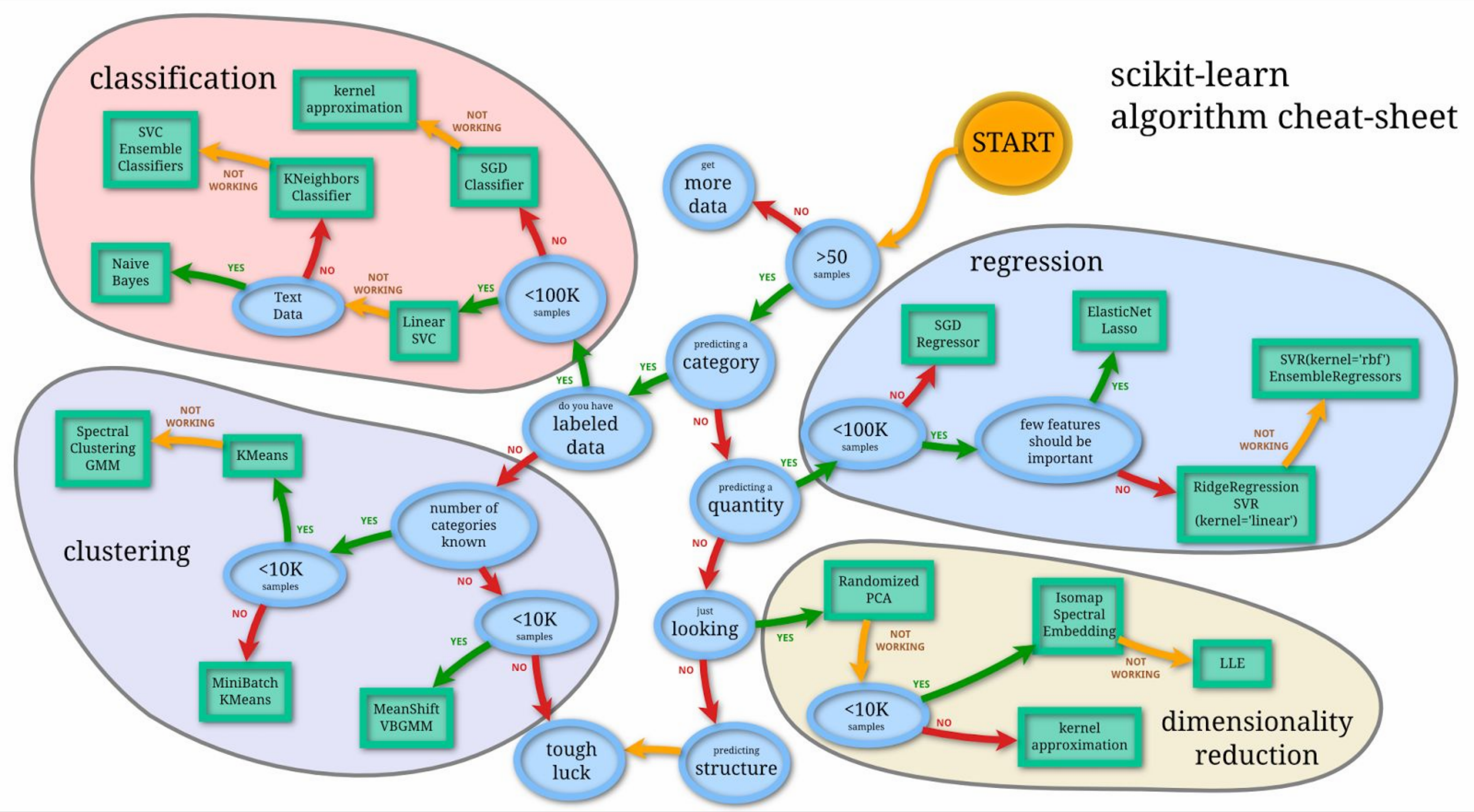


Data

- **from sklearn import datasets**
- Iris, Digits are excellent for classification
- Boston for regression
- Any classification dataset (sans labels) for clustering
- Very good for generating data

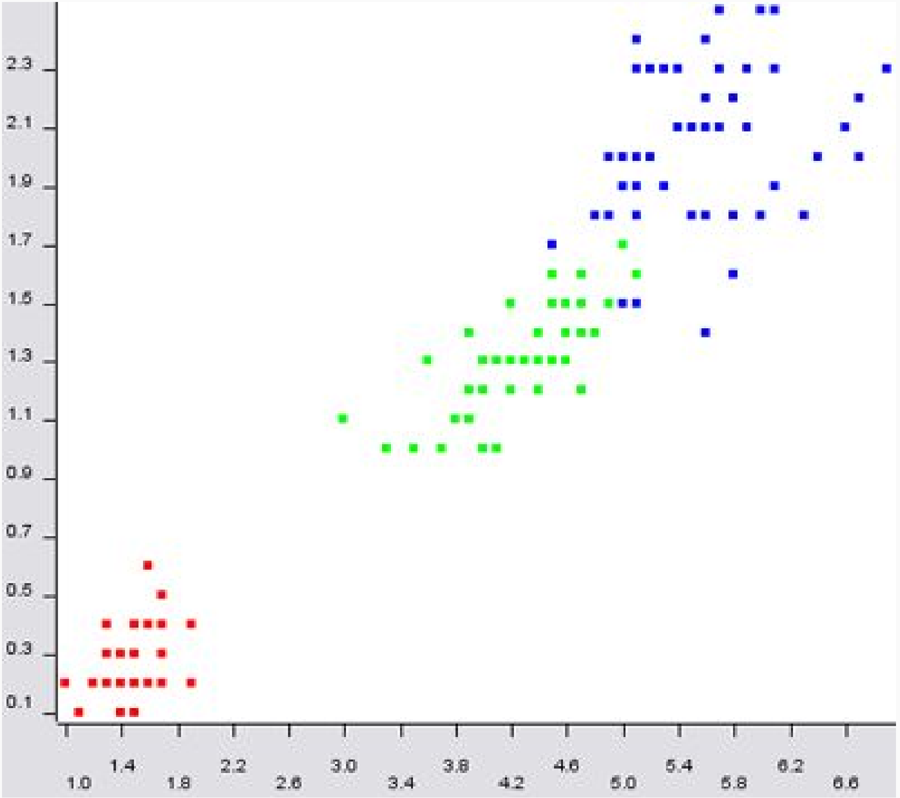


Selecting an Algorithm



The Machine Learning “World”

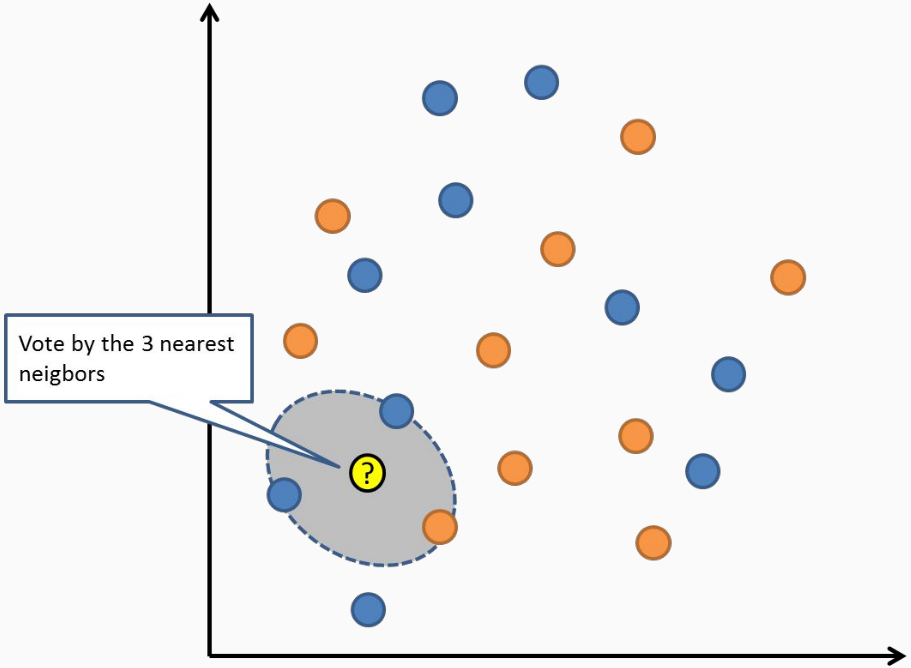
http://informationandvisualization.de/files/scatter2a.png



- Everything exists on an N dimensional cartesian plane
- Theoretically (and radically) it is possible to predict anything in the world as long as you have the right cartesian space
- And the right equation (more on this in next few slides)
- Question: How would you classify an unknown point on this diagram?

KNN (K Nearest neighbour)

- “You become who you drink coffee with”. Robin Sharma
- Simply look at the “k” nearest points around you
- Predict the same category as most of them
- Training time is zero. Since you just need to store the data and you’re done!
- Prediction time increases in O(n squared)
- Almost exclusively never used.
- Good for a starting concept.



http://3.bp.blogspot.com/-ZslDMqm5M9o/T8ja_f_fALI/AAAAAAAAAt4/z7w55YAZXpw/s1600/p1.png

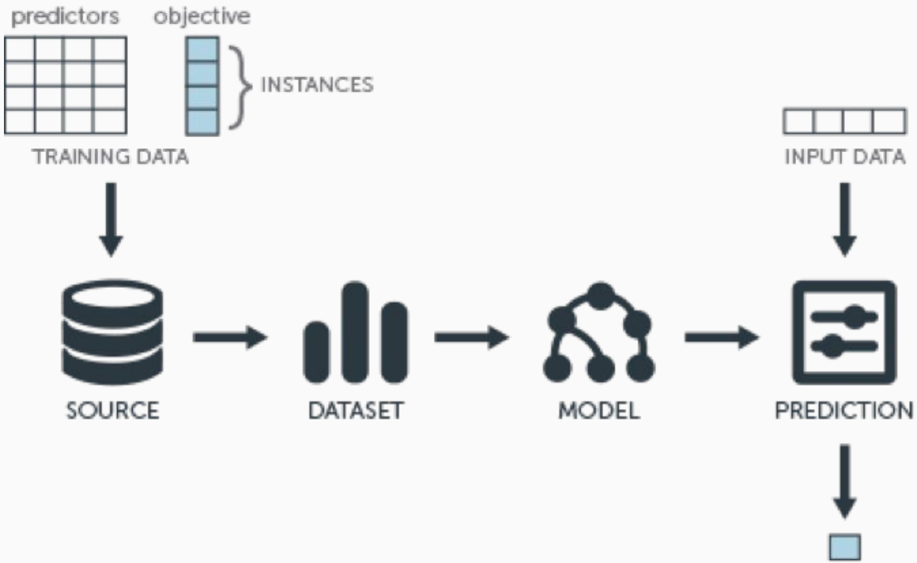
Generalizing the Idea

- Each machine learning algorithm has 2 parts
    - Training
    - Prediction

- Generally algorithms which take less time in prediction are preferred.

- The next few algorithms we discuss will have two major sub-components
    - The hypothesis function (used in making prediction)
    - The cost function (used to measure goodness of training)

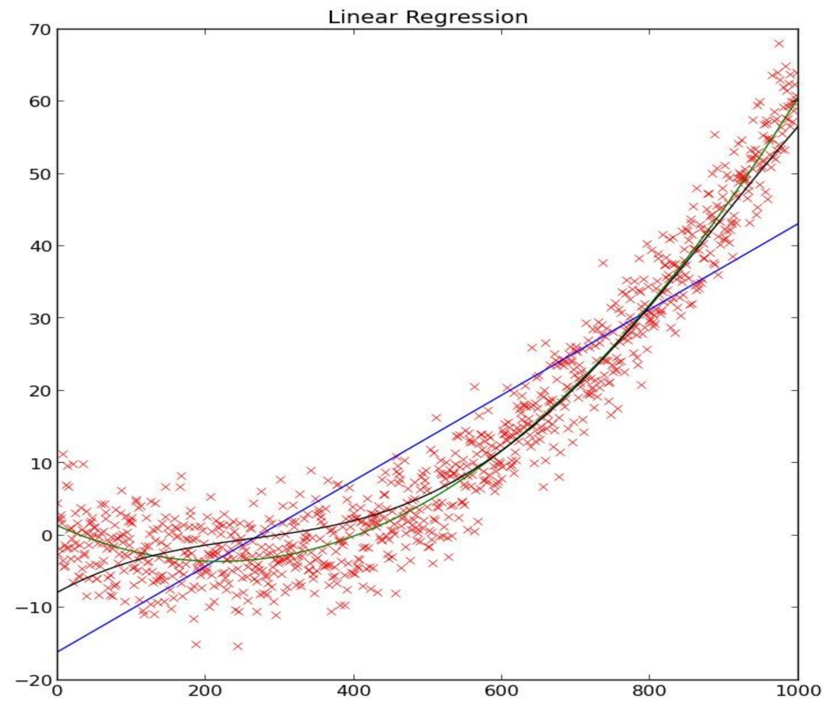
- The following cycle goes on and on until we have a reasonable model
    - Predict using current hypothesis
    - Find how good the prediction was
    - Update the hypothesis.



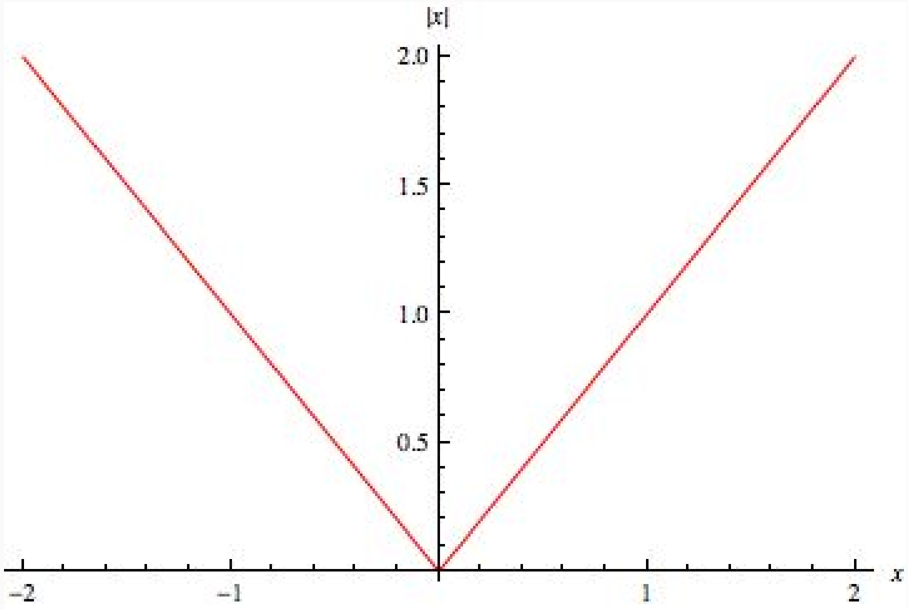
https://bbvaopen4u.com/sites/default/files/img/embed/new/cibbva_modelo.png

Linear Regression

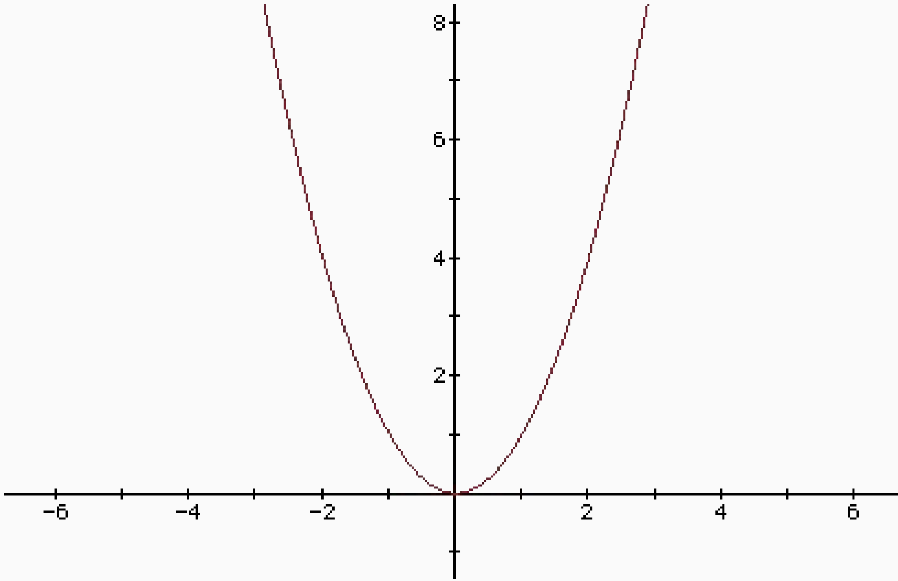
- Find the "best fit" line
- Outliers will greatly affect results
- Hypothesis function is given by
  - h(x) = w0 +w1(*x*1) + w2(*x*2) + wn(xn)
- Cost is given by sum of squared differences.
  I.e. (h(x) - y)^2



Optional Slide (Absolute error vs squared error)



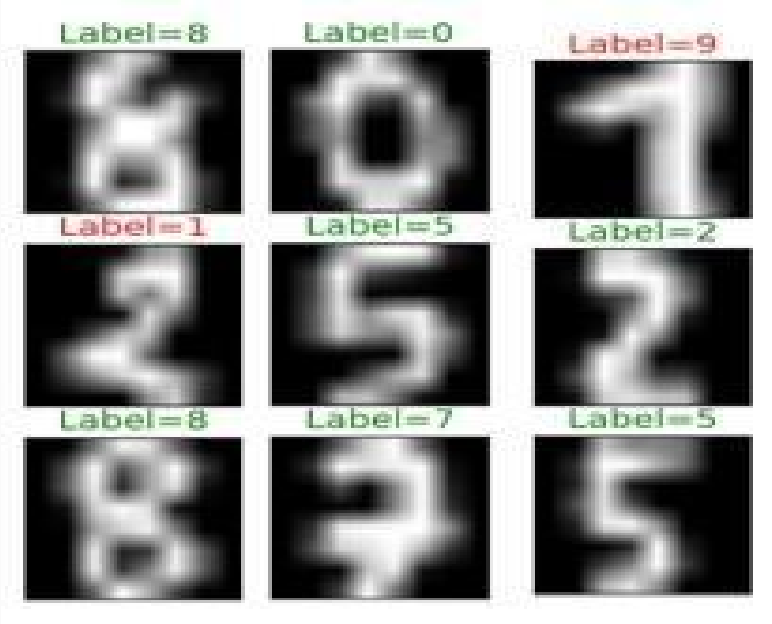
http://mathworld.wolfram.com/images/interactive/AbsReal.gif



http://jwilson.coe.uga.edu/emt668/emt668.folders.f97/wynne/Quadratic/image11.gif

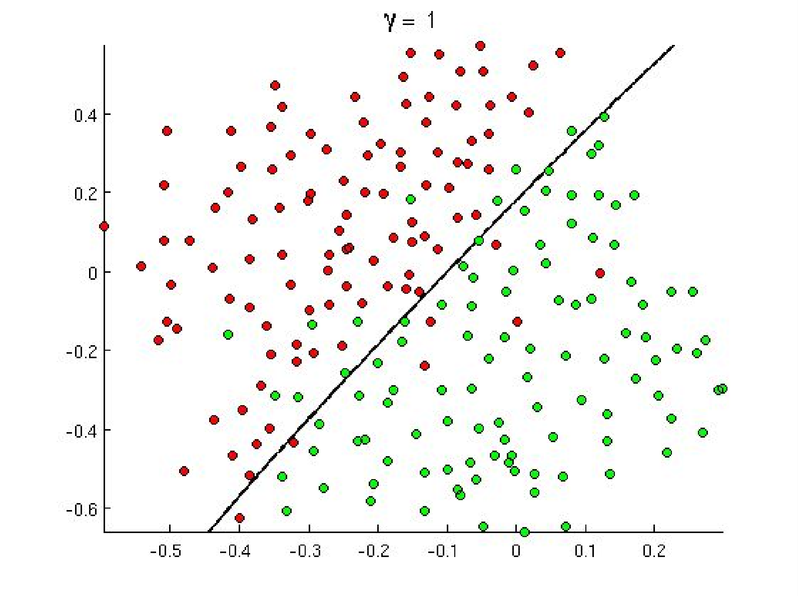
Logistic Regression

- Simple method for classification
- Uses regression to split classes
- Can be very powerful, especially after PCA (Preprocessing method)



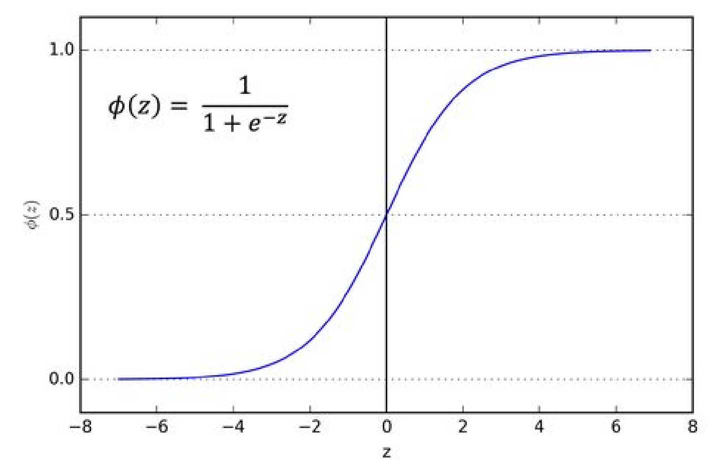
Logistic Regression

- It is a way of “classification”, unlike linear regression.
- The decision boundary may or may not be linear
- The hypothesis function goes through one additional step. The sigmoid.
- Sigmoid function returns a number between 0 and 1
- Can “assume” it to be probability.
- If sigmoid(<your function>) > 0.5:
  - predict true
  - Else predict false



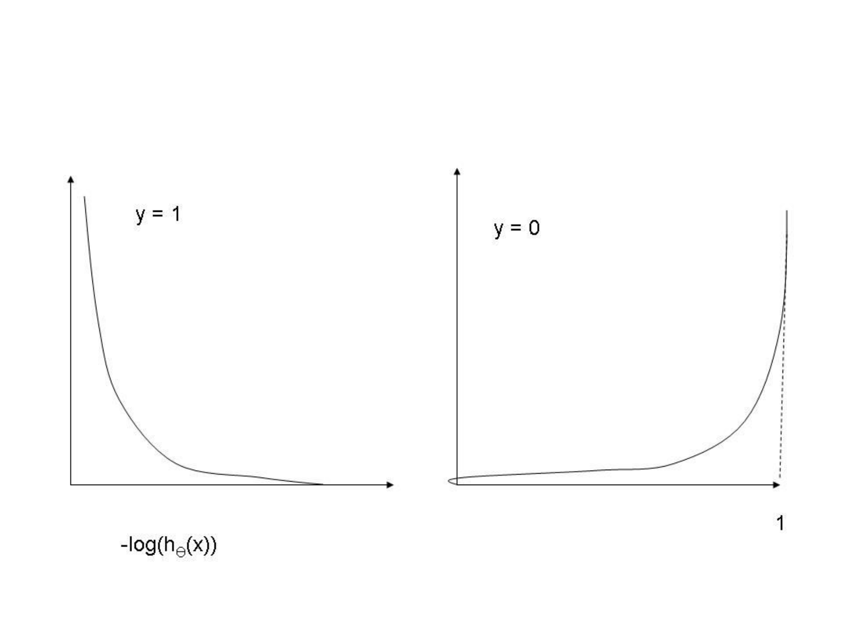
Logistic Regression (Sigmoid Function and Cost function)

http://sebastianraschka.com/images/faq/logisticregr-neuralnet/sigmoid.png



Sigmoid Function

https://gigadom.files.wordpress.com/2013/11/7.jpg

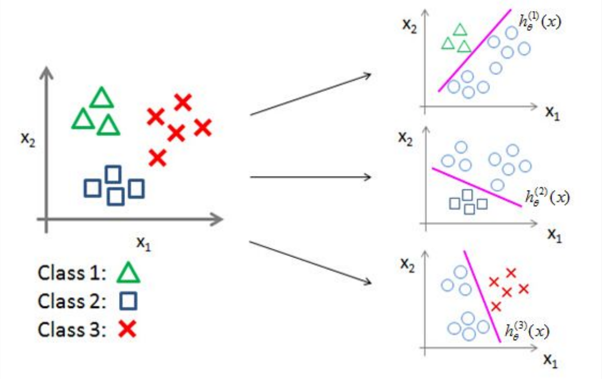


Cost Function

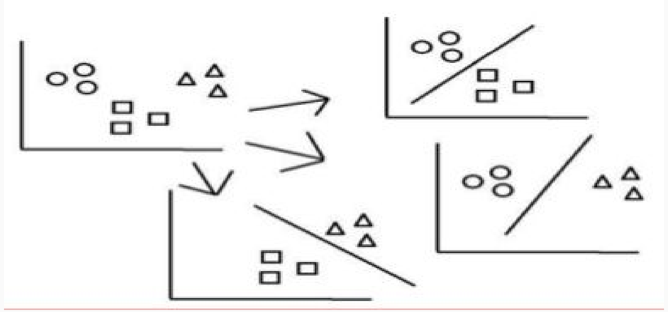
Logistic Regression (contd.)

- What if there are more than 1 classes?

- Two main approaches
  - One vs One
  - One vs All



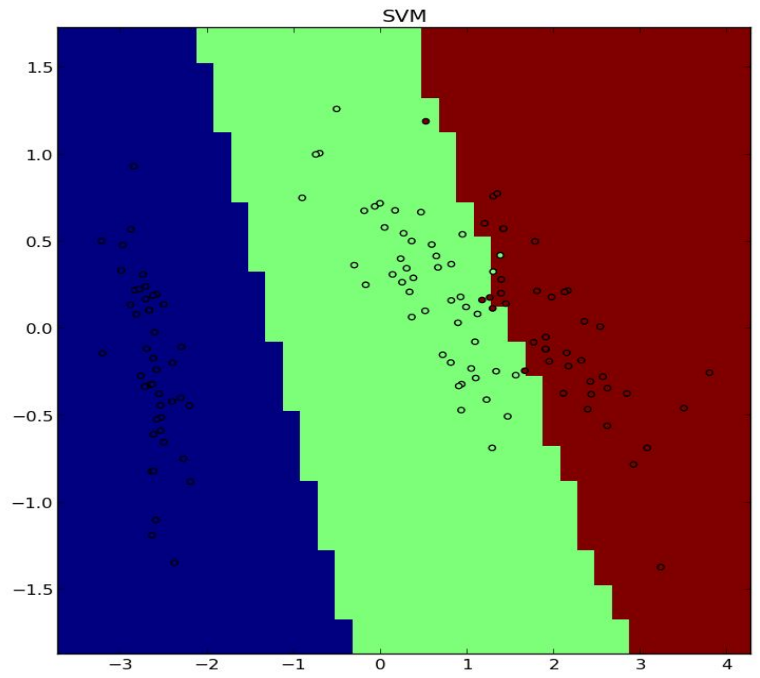
One vs All
http://img.blog.csdn.net/20160218143343043



One vs One

Support Vector Machine (SVM)

- Margin parameter is a configurable "allowed error" to account for class overlap



Some Important Points

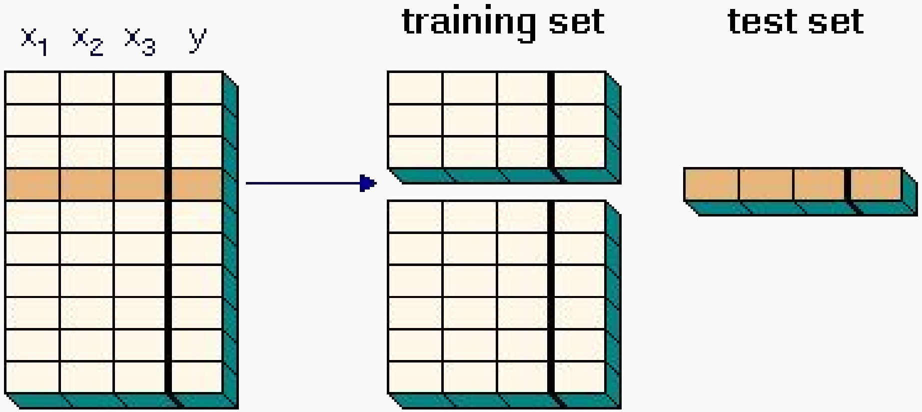
1. Cross Validation
2. Skewness of Classes (Precision vs Recall)
3. Normalization
4. Overfitting vs Underfitting
5. PCA (Optional/Superficial)

Cross Validation

- Divide your data into N chunks

- Train your algorithm N times
  - Each Time keep 1 of they chunks for testing the accuracy
  - Train on other N-1 chunks

- This is not the same as “Test Set”



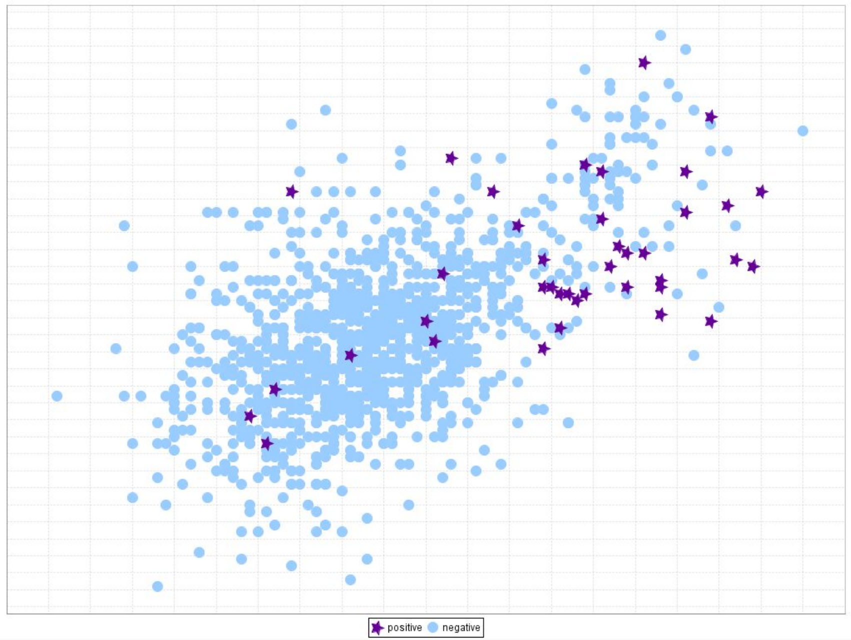
Source: http://www.statistics4u.com/fundstat_eng/img/hl_crossval.png

Skewness of Classes (Precision vs Recall)

- A tricky situation occurs when one class is over-represented in the data set.

- One way to measure performance is using the precision recall curves.
    - *Precision* describes how many of the data records, which got classified as true, actually are true.
    - *Recall* refers to the percentage of correctly classified positives of the data set.

- Various ways to reduce it
    - Limit the Over-Represented Class
    - Penalize false positives/negatives more



http://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/imbalanced/yeast4_s1.0tr_mcg_vs_gvh.png

Normalization

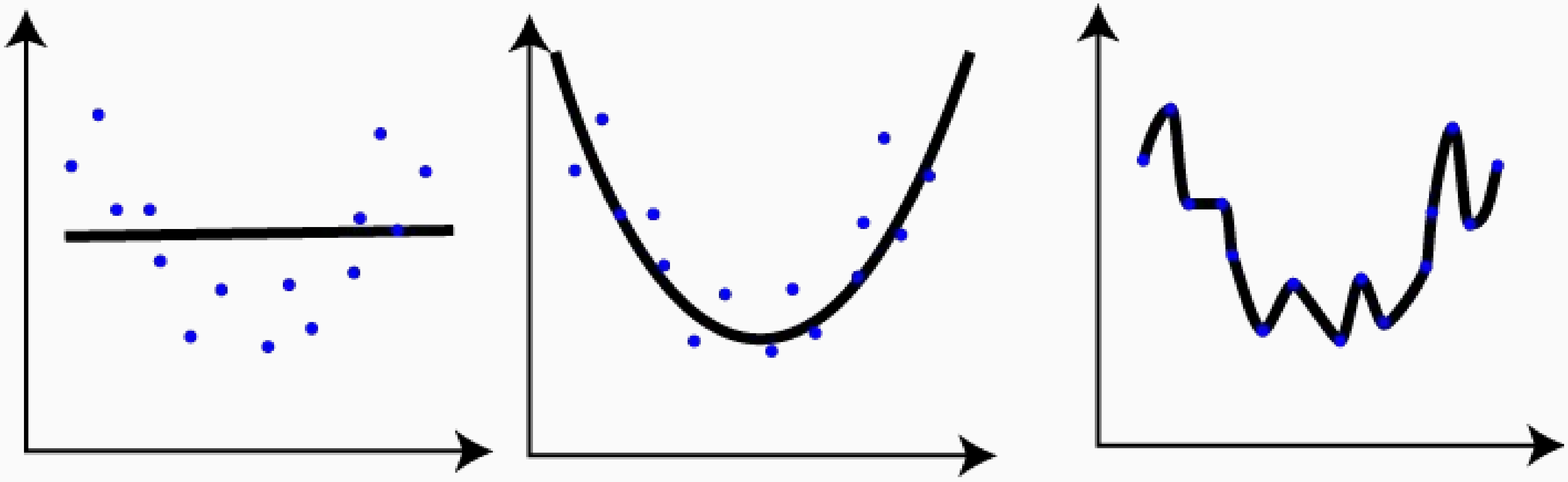
- Different features can have different range of values
- Good Idea to bring them all in same range
- Hence we use normalization/feature scaling
- For each feature f[i] in f.
  - scaled(f) = (f[i] - mean(f)) / stdev(f)
- Ensures that each feature has zero mean and unit standard deviation

Size ($feet^2$)	Price ($1000)
x	y
2104	460
1416	232
1534	315
852	178
...	...

http://img.blog.csdn.net/20160215211340879

Overfitting v.s. Underfitting

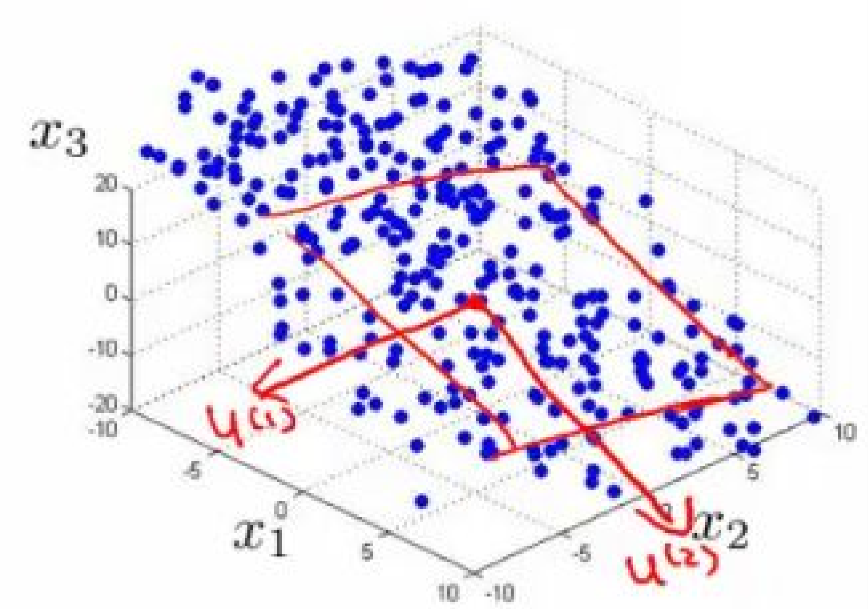
https://shapeofdata.files.wordpress.com/2013/02/overfitting.png



Underfit :(Just Right :)   Overfit :(

Dimensionality Reduction

- Having too many features for less data results in bad performance.
- Even the closest points have significantly large distances in higher dimensions!
- We could reduce the number of features we use, thereby reducing the dimensions of our data
- **Dimensionality reduction** or **dimension reduction** is the process of **reducing** the number of random variables under consideration, via obtaining a set of principal variables (Wikipedia)



http://www.holehouse.org/mlclass/14_Dimensionality_Reduction_files/Image%20[10].png

Useful Resources

Machine Learning Courses available in more depth on

- Coursera.com
- Edx.org
- Udacity.com

Any Questions? :)

Thank you!