GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

COURSE REVIEW & WHERE TO GO NEXT

I. WHERE HAVE WE BEEN?

REMEMBER THIS...?

	continuous	***categorical***
supervised	*regression*	*classification*
unsupervised	*dimension reduction*	*clustering*

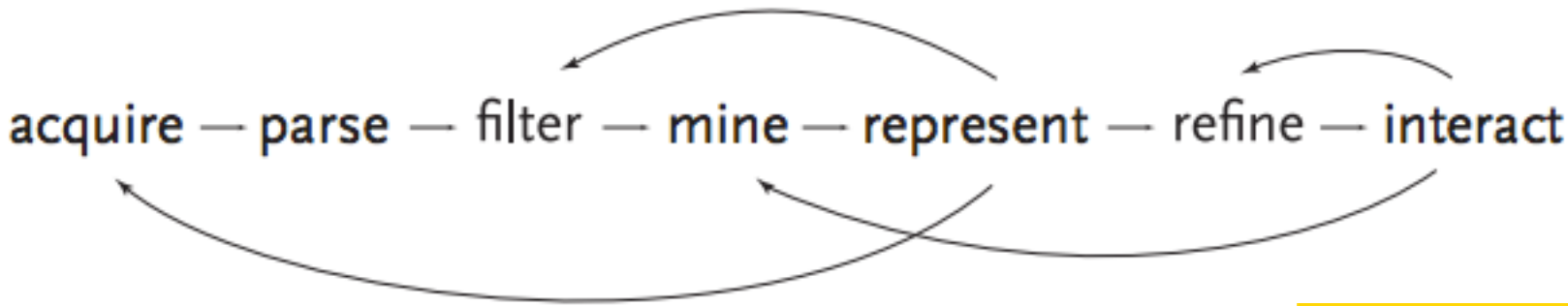
WE COVERED

- *Data acquisition and preparation*
- *Exploratory data analysis*
- *Supervised learning:*
  - *kNN*
  - *Linear, multiple, & polynomial regression*
  - *Decision trees & random forests*
  - *logistic regression*
  - *Naïve Bayes*
  - *Support Vector Machines (SVM)*

WE COVERED

- *Unsupervised learning:*
  - *K-means clustering*
  - *PCA/SVD for dimensionality reduction*
- *Model evaluation, confusion matrix, ROC curves, AUC*
- *APIs & web scraping*
- *Recommender systems*
- *Databases and SQL*
- *Text mining and natural language processing*

THE DATA SCIENCE WORKFLOW



NOTE

This diagram illustrates the *iterative* nature of problem solving

source: http://benfry.com/phd/dissertation-110323c.pdf

GIT COMMANDS

Main

git clone – clone a repo
git status – get status
git add – add changes to be pushed
git commit – commit the change with a comment
git push – push the change to github
git pull – pull remote changes from github

Others

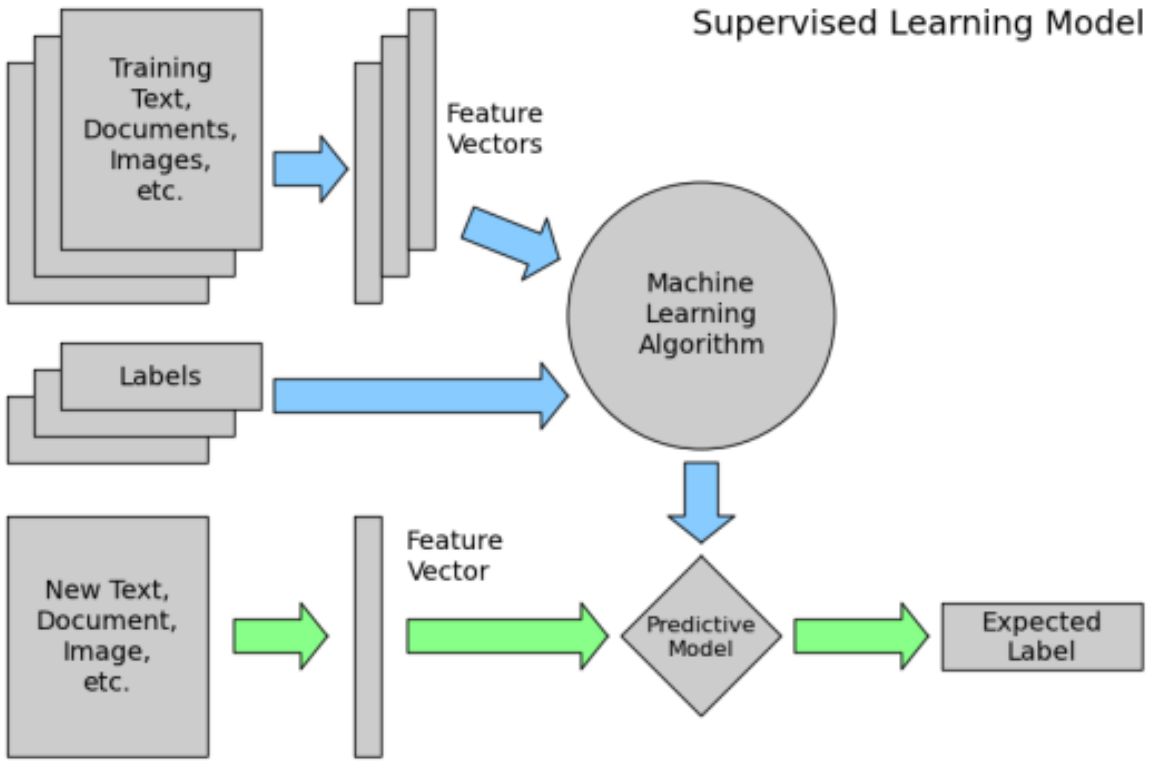
git branch – see all branches
git checkout – checkout a branch
git merge – merge in another branch
git stash – stash changes
pull request – remote changes requested to be merged in

PEP 20: THE ZEN OF PYTHON

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
```

http://legacy.python.org/dev/peps/pep-0020/

How does supervised learning “work”?

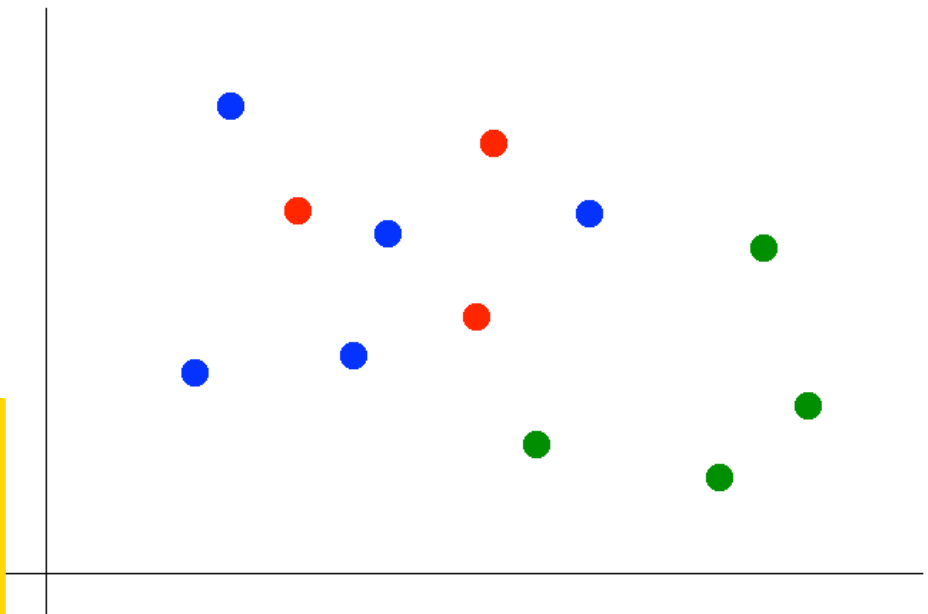


Suppose we want to predict the color of the gray dot.

1) *Pick a value for k.*

2) *Find colors of k nearest neighbors.*

3) *Assign the most common color to the gray dot.*



NOTE:

Our definition of "nearest" implicitly uses the *Euclidean distance function.*

OVERFITTING & UNDERFITTING

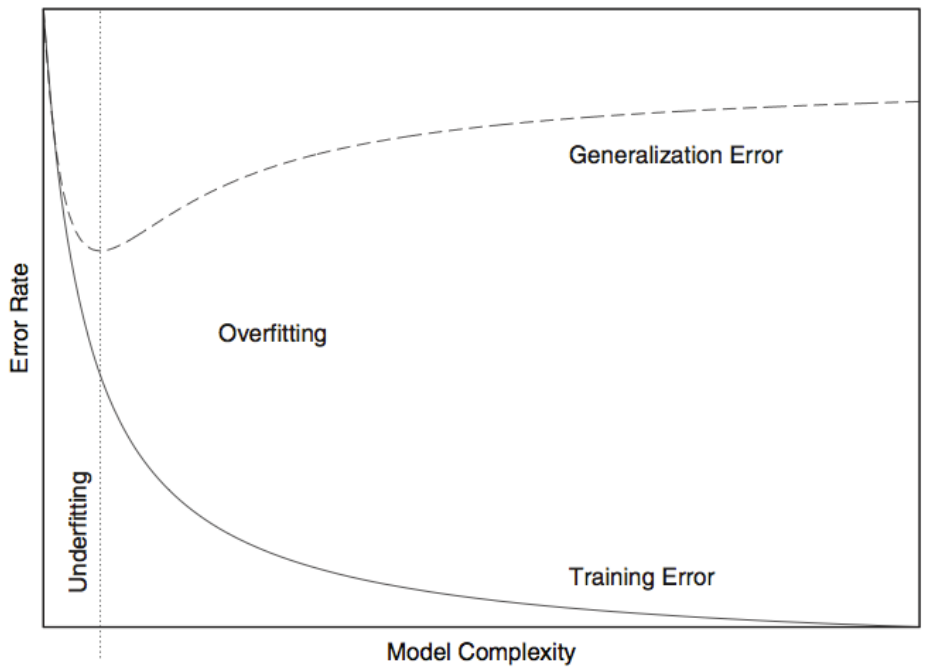
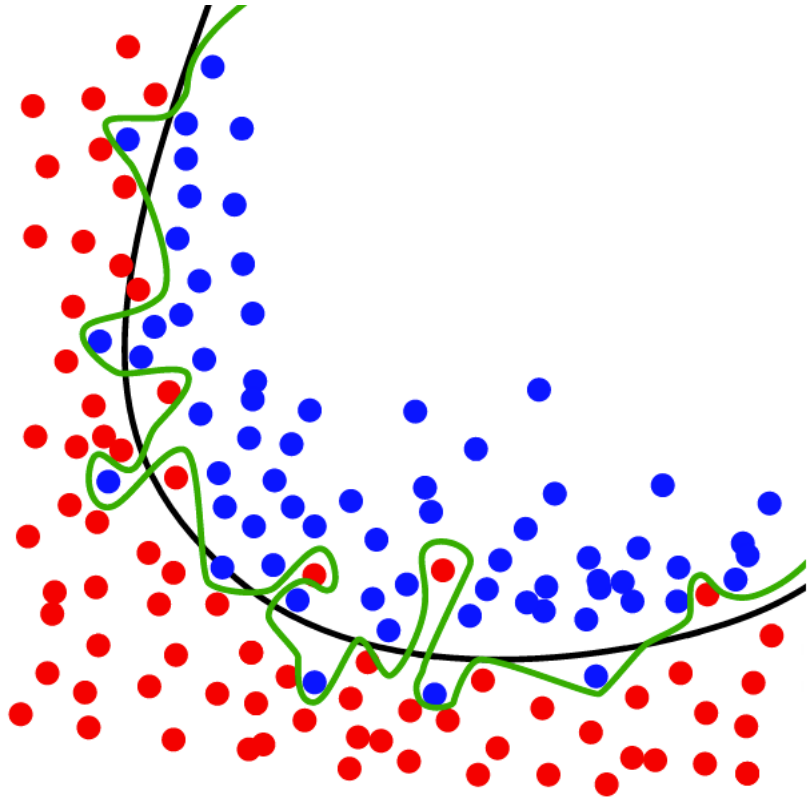


FIGURE 18-1. Overfitting: as a model becomes more complex, it becomes increasingly able to represent the training *data. However, such a model is overfitted and will not generalize well to data that was not used during training.*

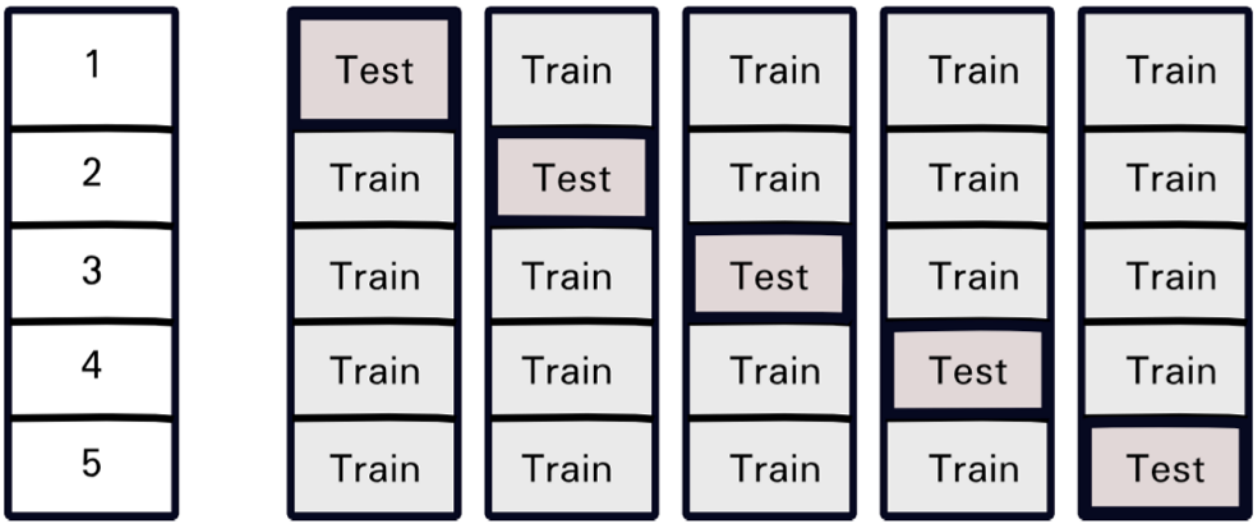
source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.

OVERFITTING - EXAMPLE



source: http://www.dtreg.com

CROSS-VALIDATION: 5-FOLD EXAMPLE



Accuracy

k_1 %

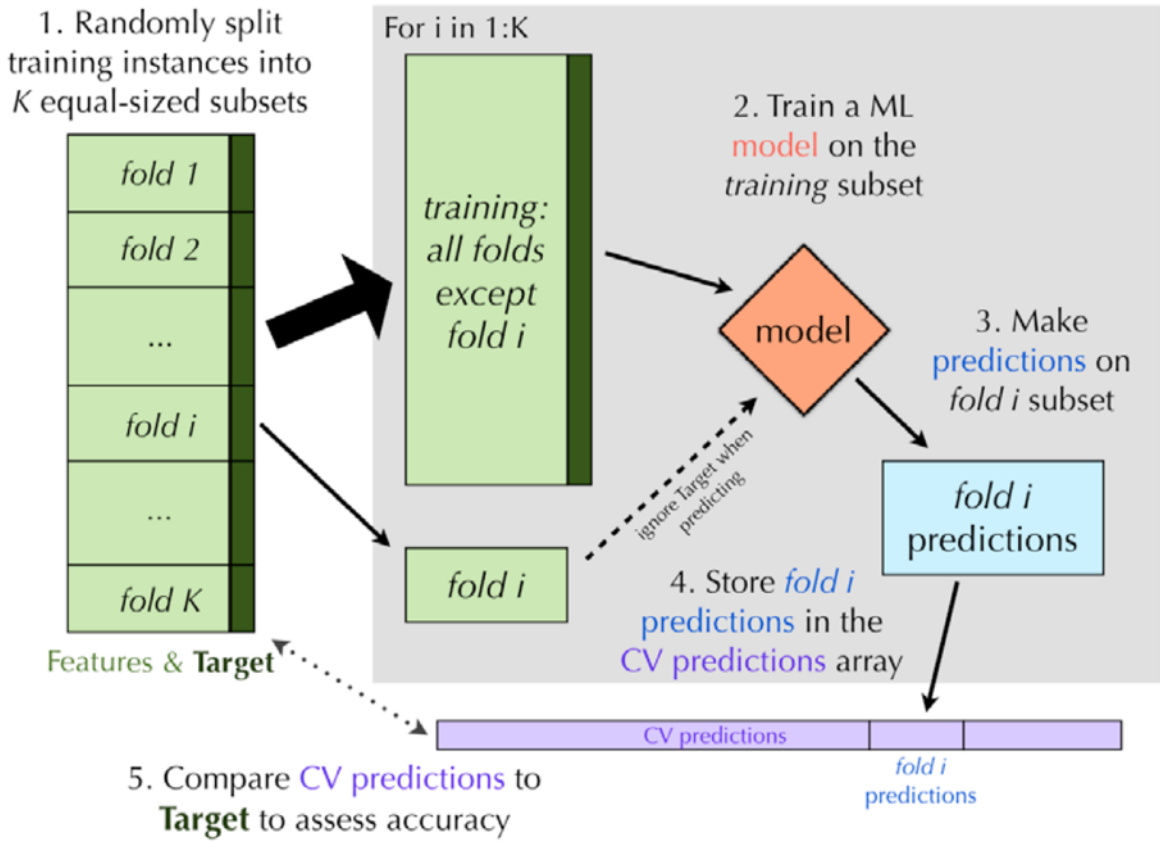
k_2 %

k_3 %

k_4 %

k_5 %

5-Fold Generalization Error = $(k_1 + k_2 + k_3 + k_4 + k_5) / 5$

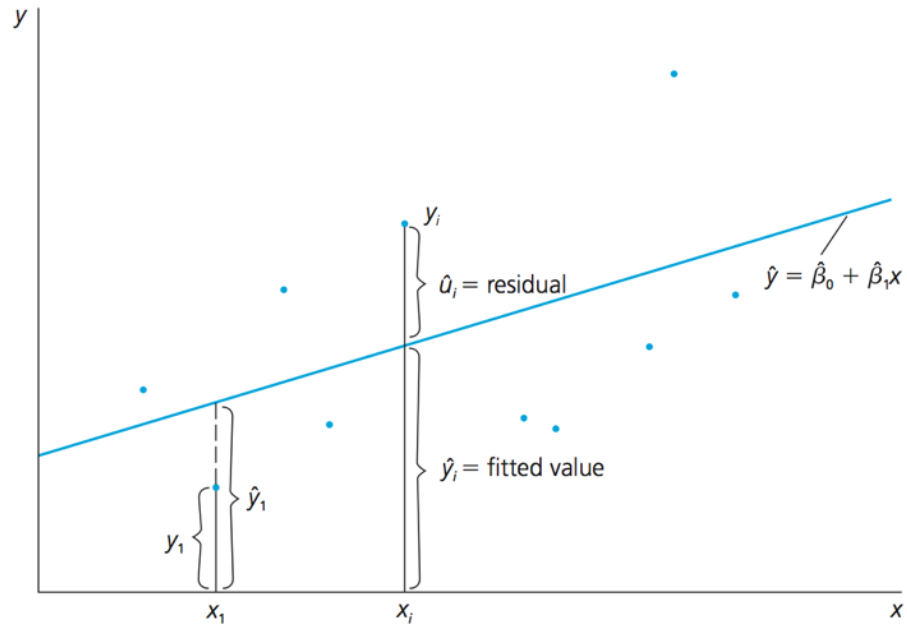

1. Randomly split training instances into K equal-sized subsets
fold 1
fold 2
...
fold i
...
fold K
Features & Target
For i in 1:K
training: all folds except fold i
fold i
2. Train a ML model on the training subset
model
ignore Target when predicting
3. Make predictions on fold i subset
fold i predictions
4. Store fold i predictions in the CV predictions array
CV predictions
fold i predictions
5. Compare CV predictions to Target to assess accuracy

Q: How do we fit a regression model to a dataset?
A: In theory, minimize the sum of the squared residuals (OLS).

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

$$\hat{u}_i = y_i - \hat{y}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i.$$

$$\sum_{i=1}^{n} \hat{u}_i^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2,$$



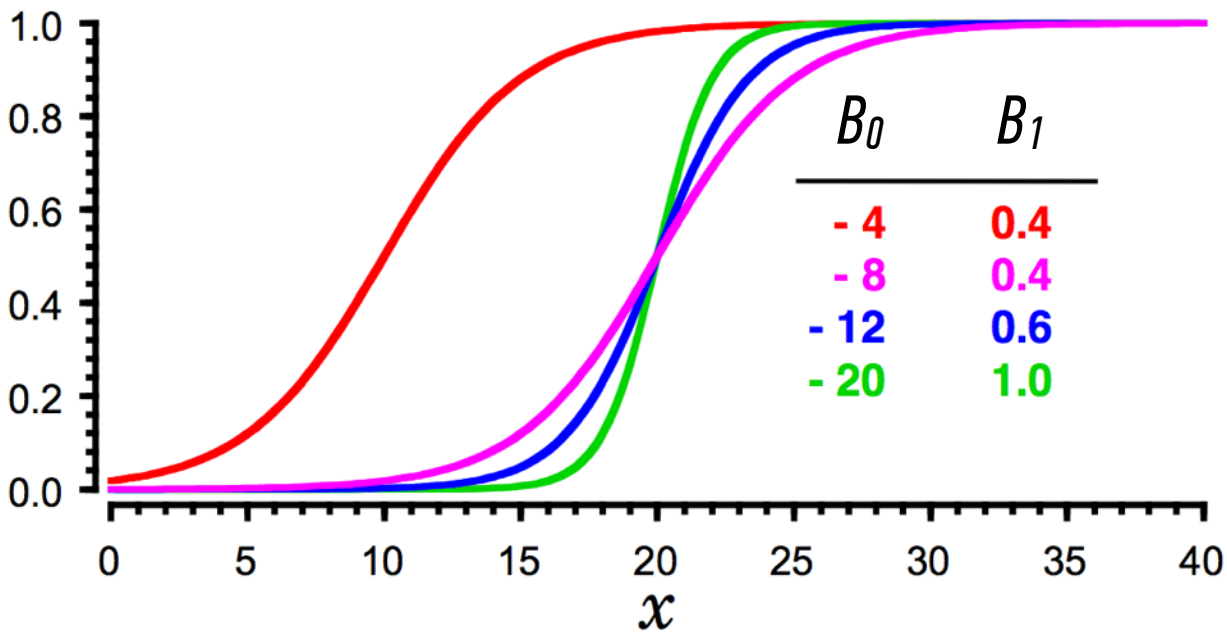
This tradeoff is regulated by a **hyperparameter** λ*, which we've already seen:*

L1 regularization*:* $y = \Sigma \beta_i x_i + \varepsilon \quad st. \quad \Sigma |\beta_i| < \lambda$

L2 regularization*:* $y = \Sigma \beta_i x_i + \varepsilon \quad st. \quad \Sigma \beta_i^2 < \lambda$

So regularization represents a method to trade away some variance for a little bias in our model, thus achieving a better overall fit.

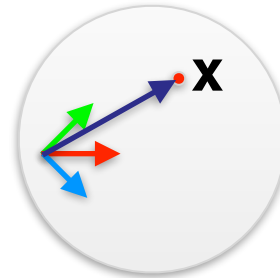
$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$



When $B_0 + B_1x = 0$, then $F(x) = 0.5$, which is the inflection point on all these curves.

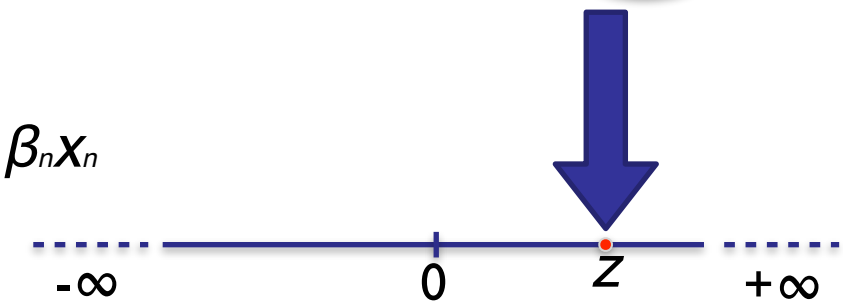
1. Model consists of a vector **β** in n-dimensional feature space

$$\boldsymbol{\beta} = \beta_1 + \beta_2 + \ldots + \beta_n$$



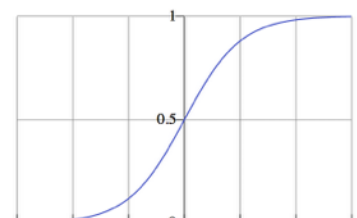
2. For a point **x**, project it onto β to convert it into a real number z in the range -∞ to +∞

$$z = \alpha + \boldsymbol{\beta}.\mathbf{x} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$



3. Map z to the range 0 to 1 using the logistic function

$$p = 1/(1+e^{-z})$$



source: http://courses.washington.edu/css490/2012.Winter/lecture_slides/05b_logistic_regression.pdf

Notice if we take the logarithm of the odds, we return a linear equation

$$\log(\frac{\pi}{1-\pi}) = \log(e^{\beta_0+\beta_1 x}) = \beta_0 + \beta_1 x$$

This simple relationship between the odds ratio and the parameter β is what makes logistic regression such a powerful tool.

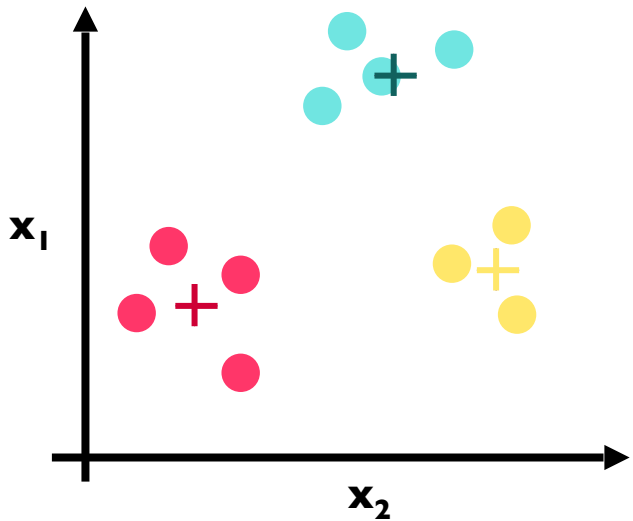
This term is the **posterior probability** *of* C*. It represents the probability of a record belonging to class* C *after the data is taken into account.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

The goal of any Bayesian computation is to find ("learn") the posterior distribution of a particular variable.

1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*



3) *recalculate centroid positions*

4) repeat steps 2-3 until stopping criteria met

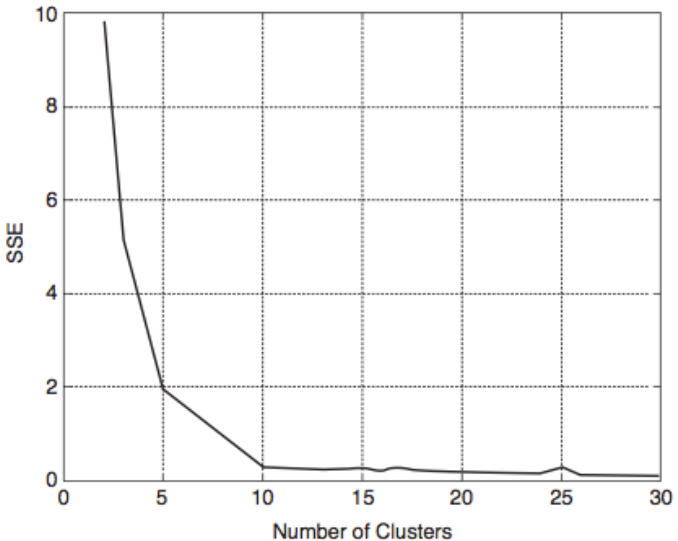


Figure 8.32. SSE versus number of clusters for the data of Figure 8.29.

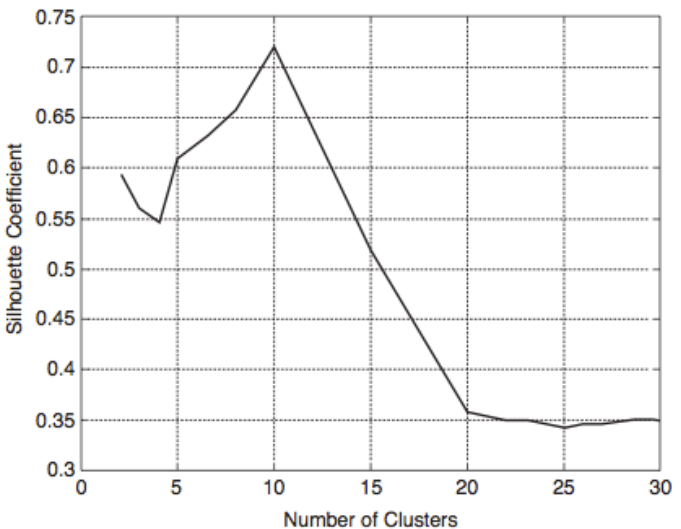
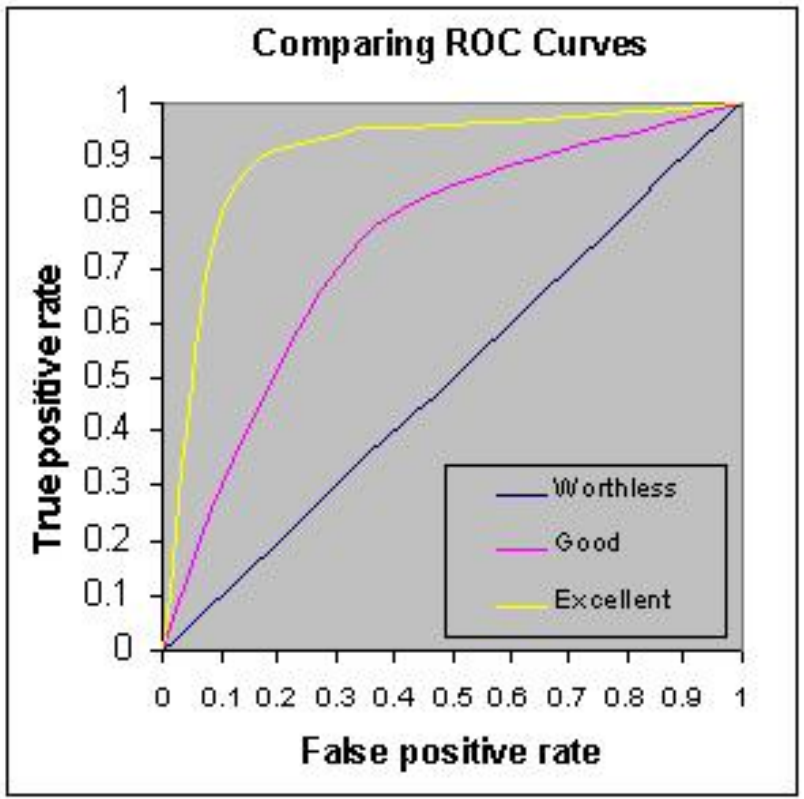
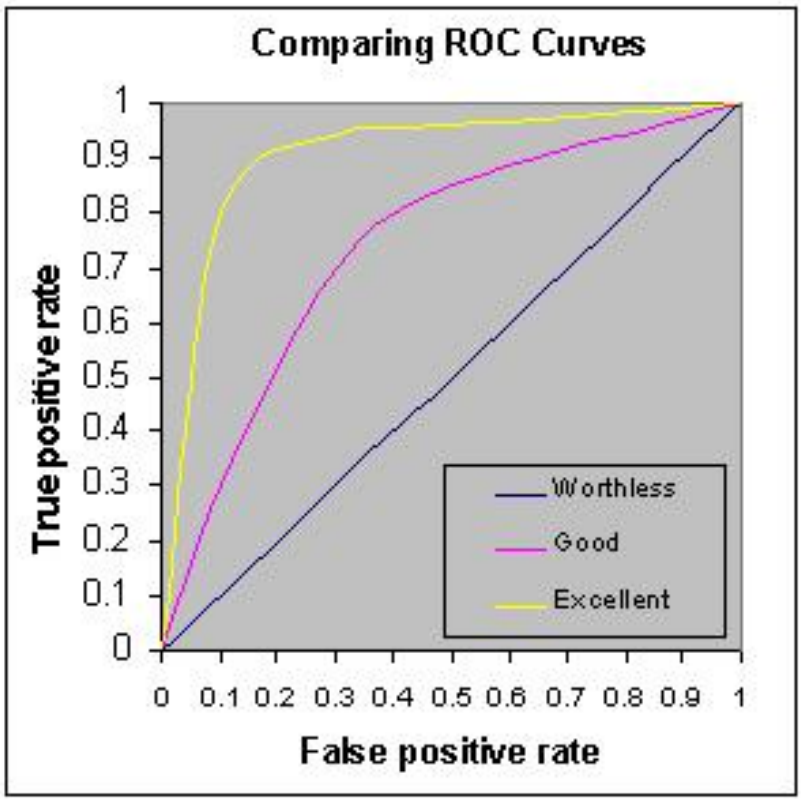


Figure 8.33. Average silhouette coefficient versus number of clusters for the data of Figure 8.29.



ROC Curves show the relationship between the TP Rate and the FP Rate as we vary the decision threshold for the classifier

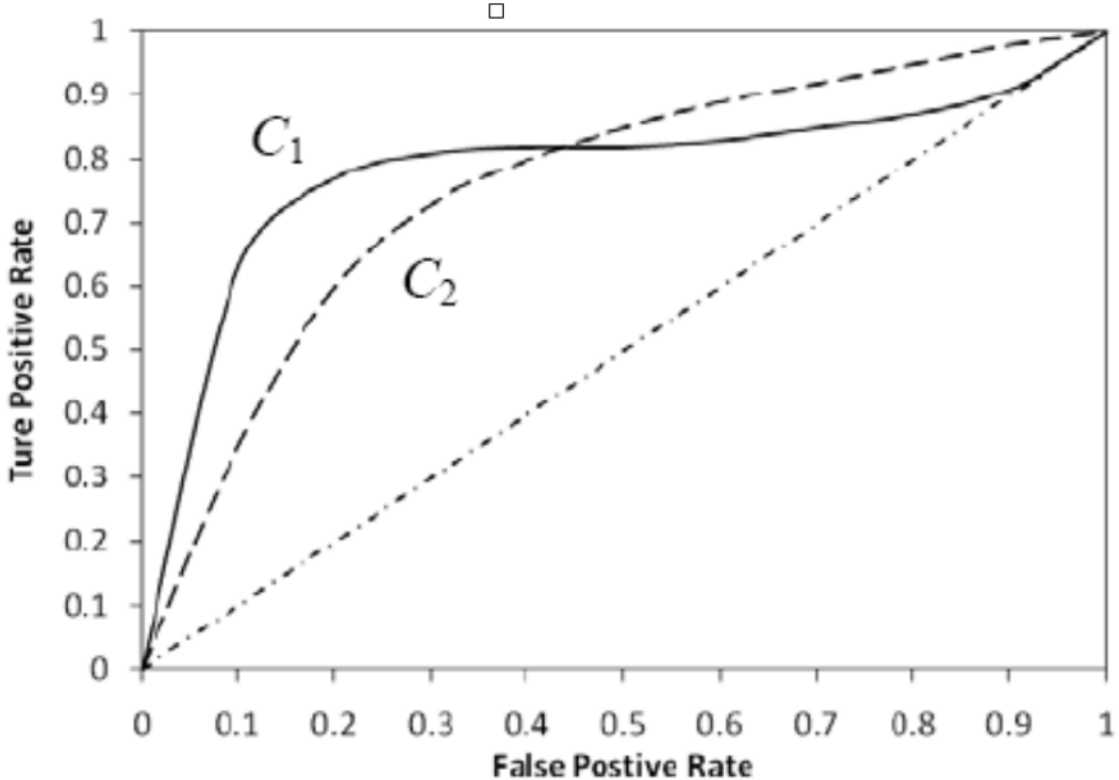
Cut off	TPR (y)	FPR (x)	Cut off	TPR (y)	FPR (x)
0	1	1	0.50	0.75	0.25
0.05	1	0.75	0.65	0.5	0
0.15	1	0.5	0.85	0.25	0
0.25	1	0.25	1	0	0



Area Under the Curve (AUC)

We evaluate a classifier by measuring the Area Under the Curve for its ROC curve. The Greater area under the curve, the more effective the classifier.

Then for our chosen classifer, we pick an appropriate decision threshold. In general, we pick the decision threshold that gets us closest to the upper left corner



Pop Quiz!

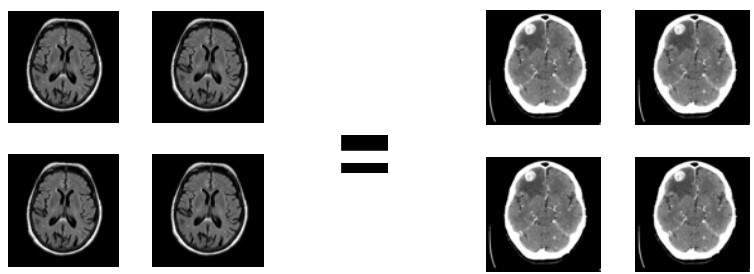
Which of the two classifiers shown (on the same data), C_1 or C_2, is better and why?

Imbalanced classes can be re-balanced in several ways

1. **Undersampling** the dominant class - remove some the majority class so it has less weight

2. **Oversampling** the minority class - add more of the minority class so it has more weight.

3. **Hybrid** - doing both



We want our objective function to measure the gain in purity from a particular split.

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

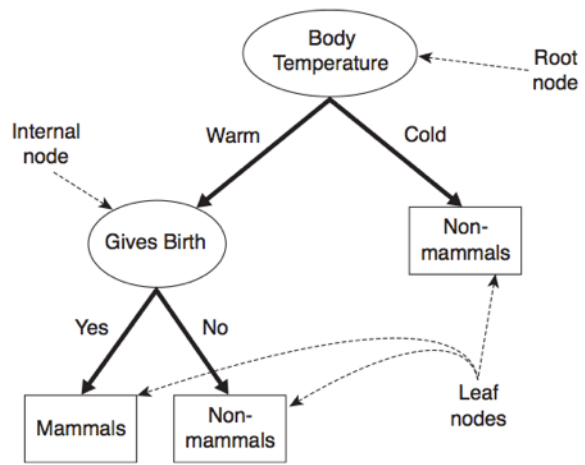
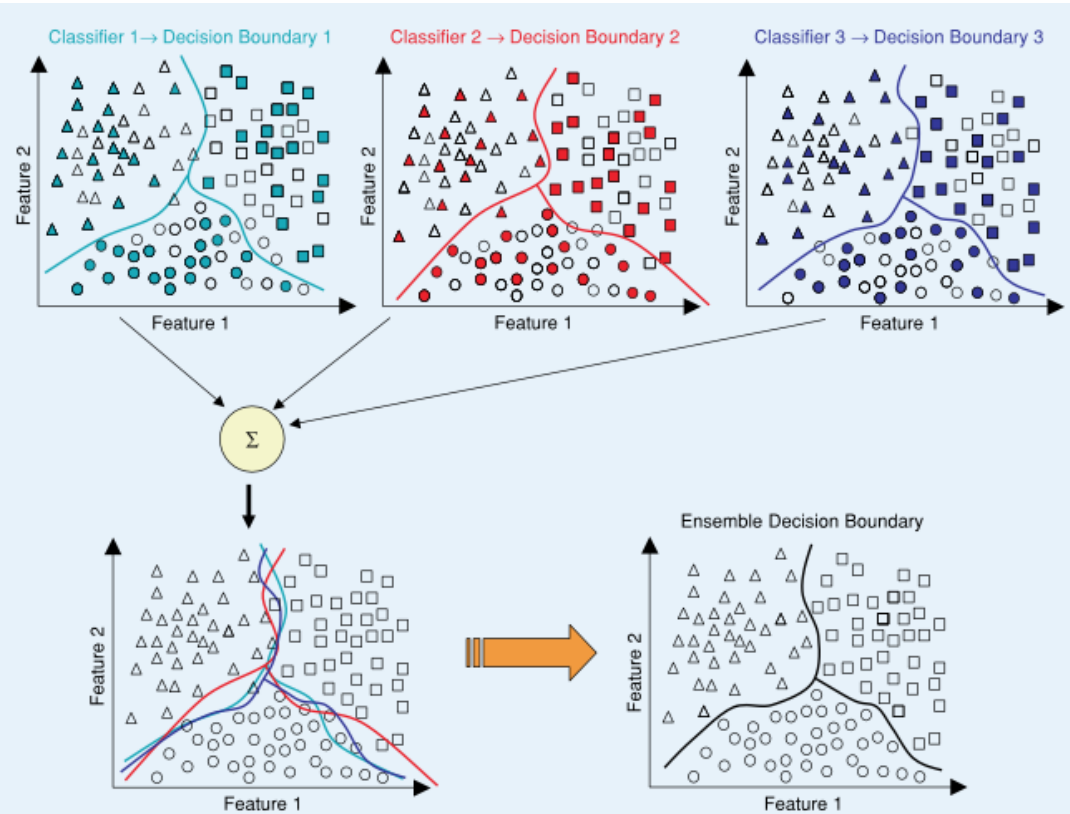


Figure 4.4. A decision tree for the mammal classification problem.


Classifier 1→ Decision Boundary 1
Classifier 2 → Decision Boundary 2
Classifier 3 → Decision Boundary 3
Feature 2
Feature 1
Σ
Ensemble Decision Boundary

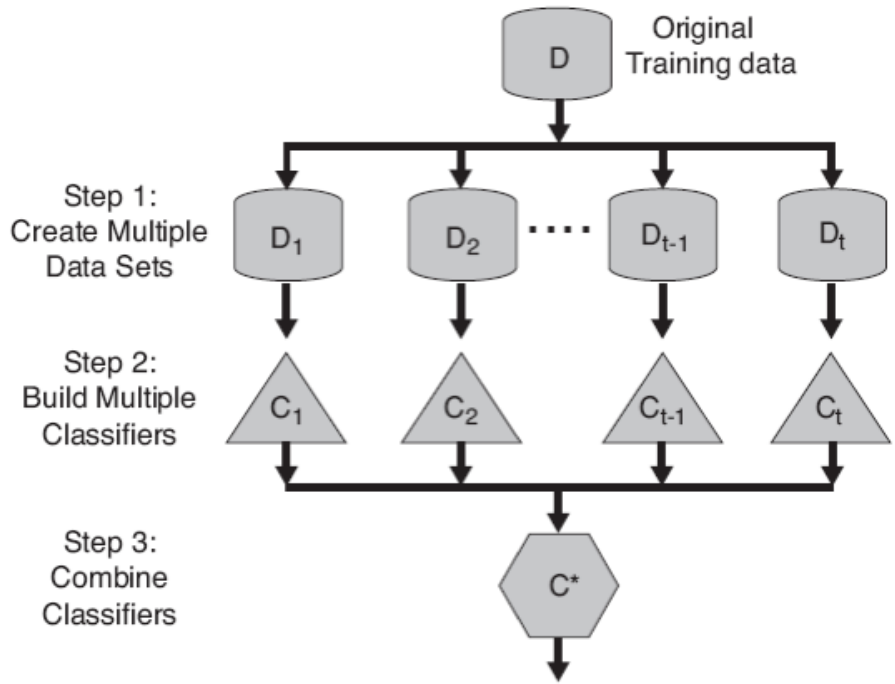
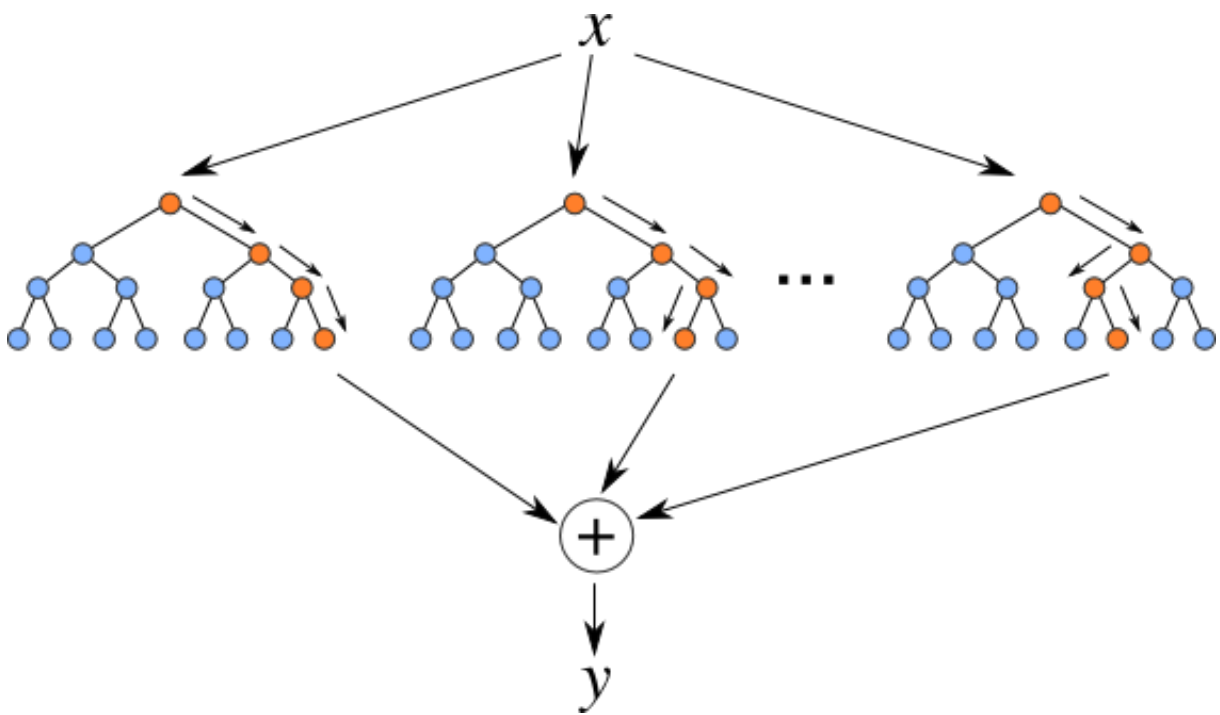
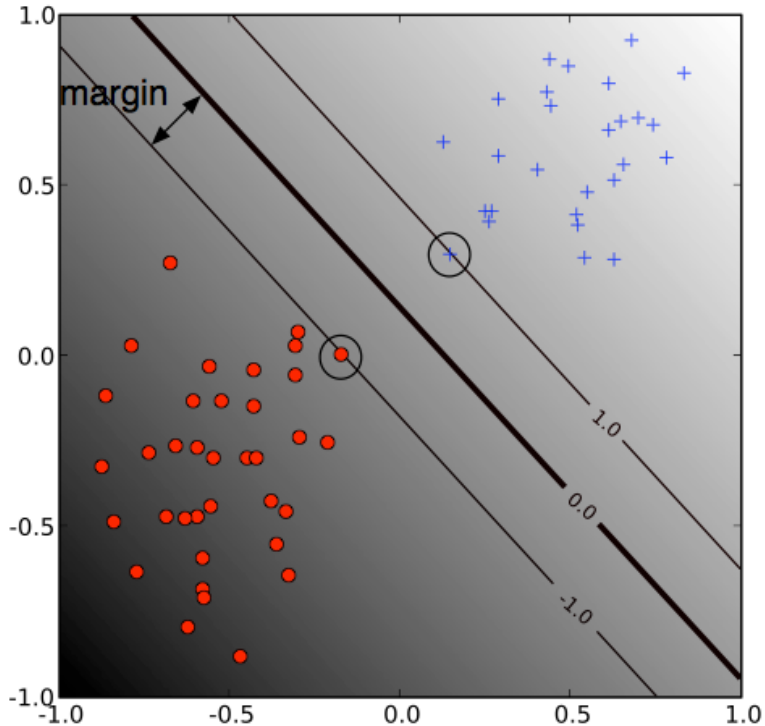


Figure 5.31. A logical view of the ensemble learning method.

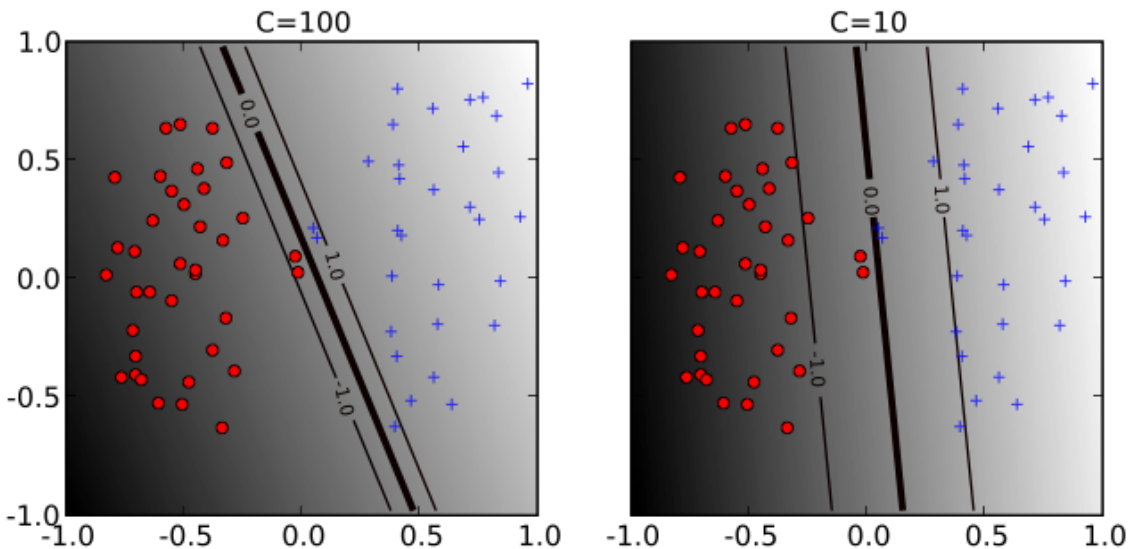

x
...
+
y

Notice that the margin depends only on the points that are nearest to the decision boundary.

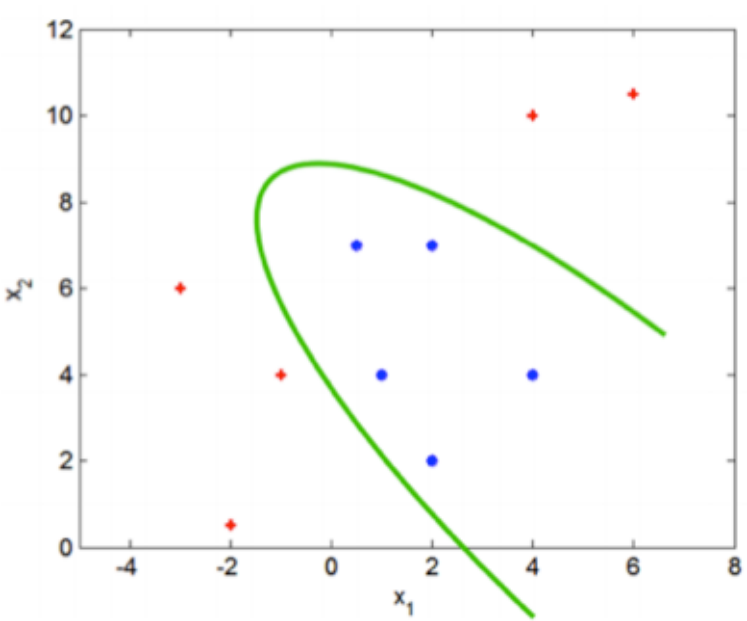
These points are called the **support vectors**.

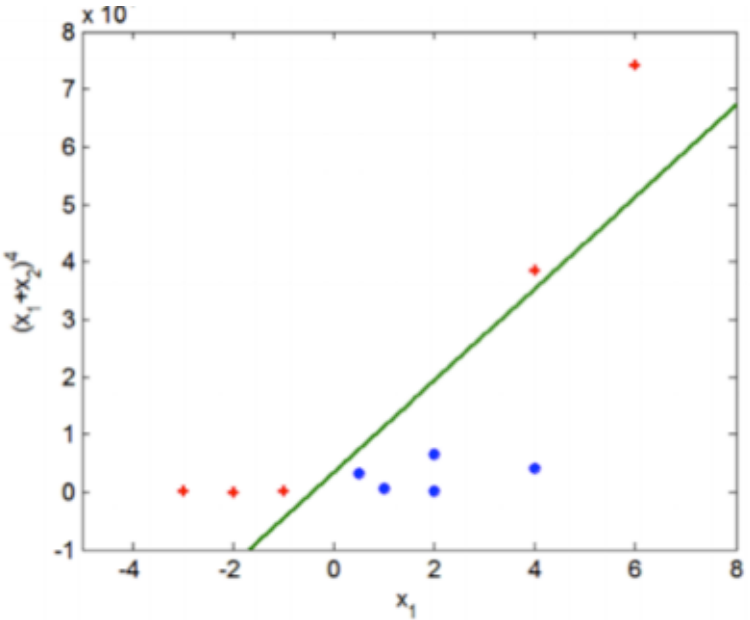


We can trade some training error in order to get a larger margin (remember the bias-variance trade-off)



source: http://pyml.sourceforge.net/doc/howto.pdf

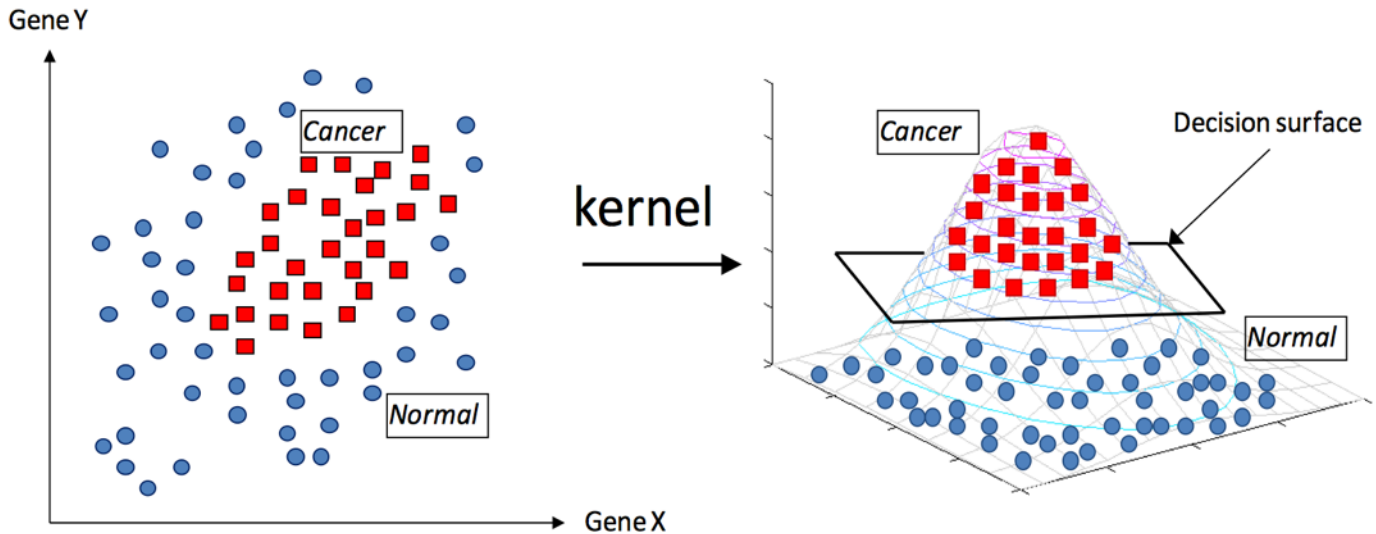




original feature space *K*

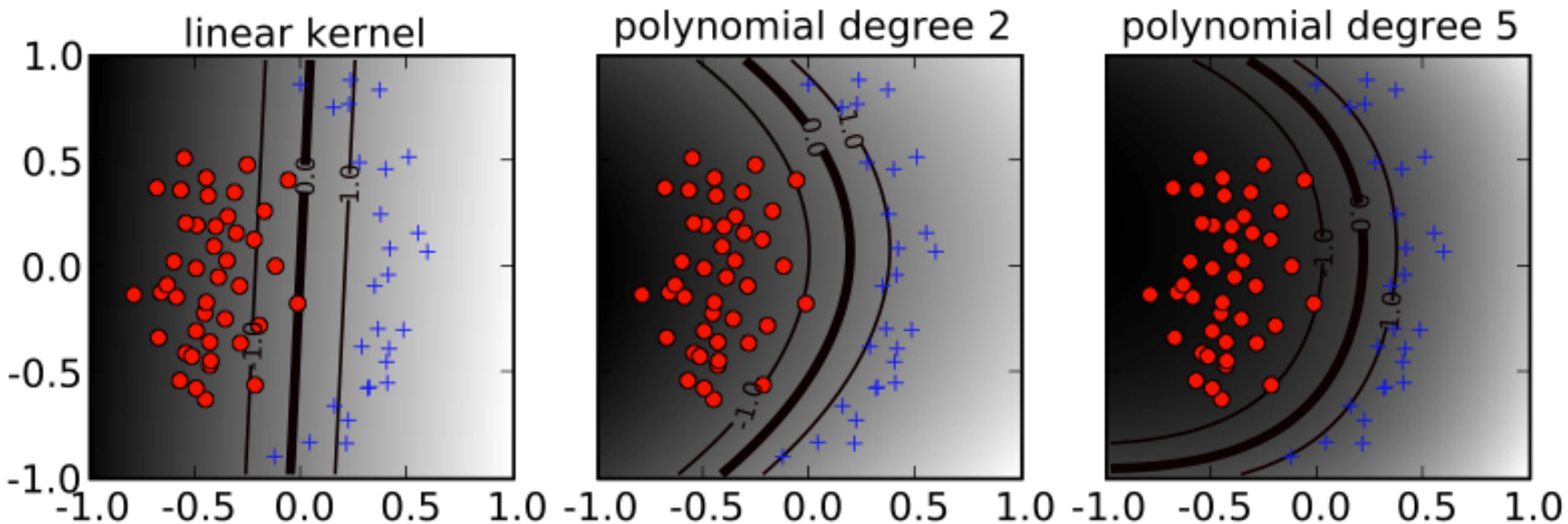
higher-dim feature space *K'*

SUPPORT VECTOR MACHINES: THE KERNEL TRICK

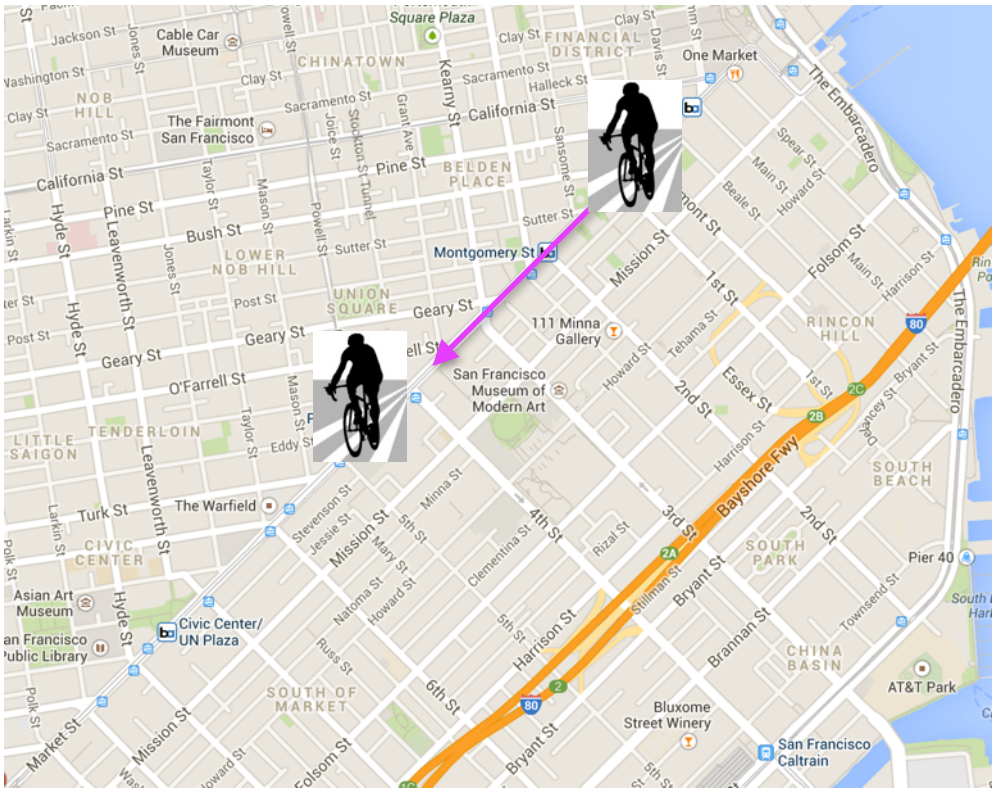


If a linear decision boundary cannot be found in the original space, we can map into a higher dimensional space and find the separating surface.

SUPPORT VECTOR MACHINES: NONLINEAR CLASSIFICATION – POLYNOMIAL KERNEL

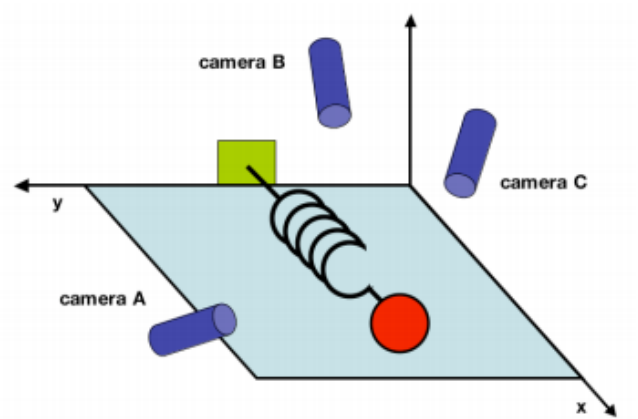


source: http://pyml.sourceforge.net/doc/howto.pdf



dist = 5

What if we just used distance down Market St.?



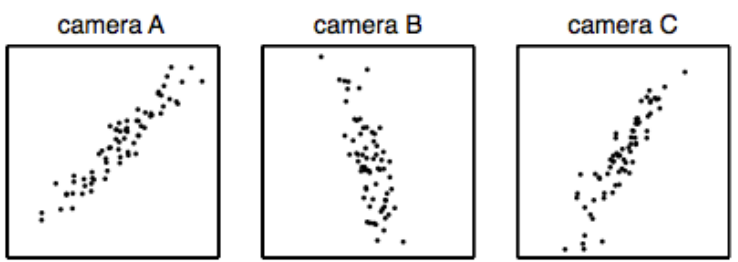
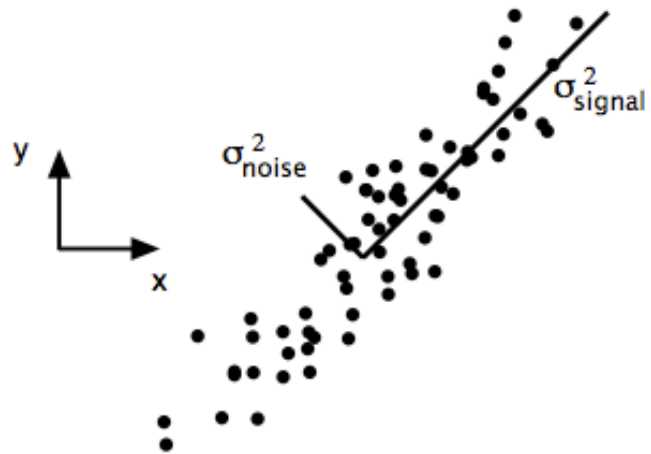


FIG. 1 A toy example. The position of a ball attached to an oscillating spring is recorded using three cameras A, B and C. The position of the ball tracked by each camera is depicted in each panel below.

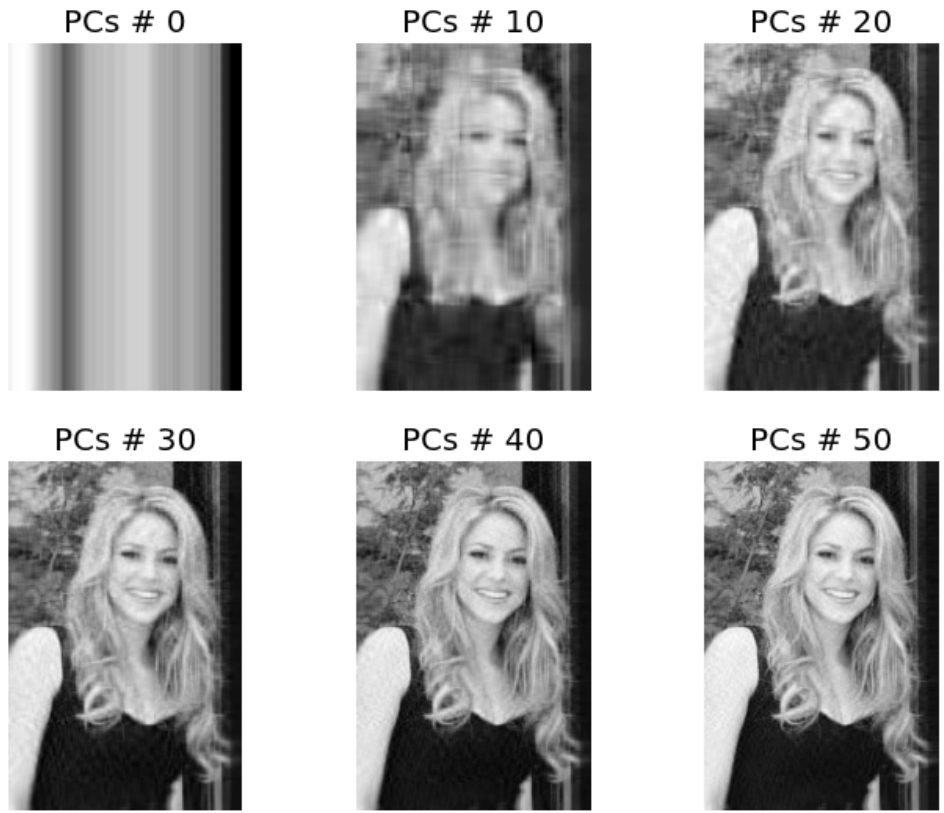
source: http://www.snl.salk.edu/~shlens/pca.pdf



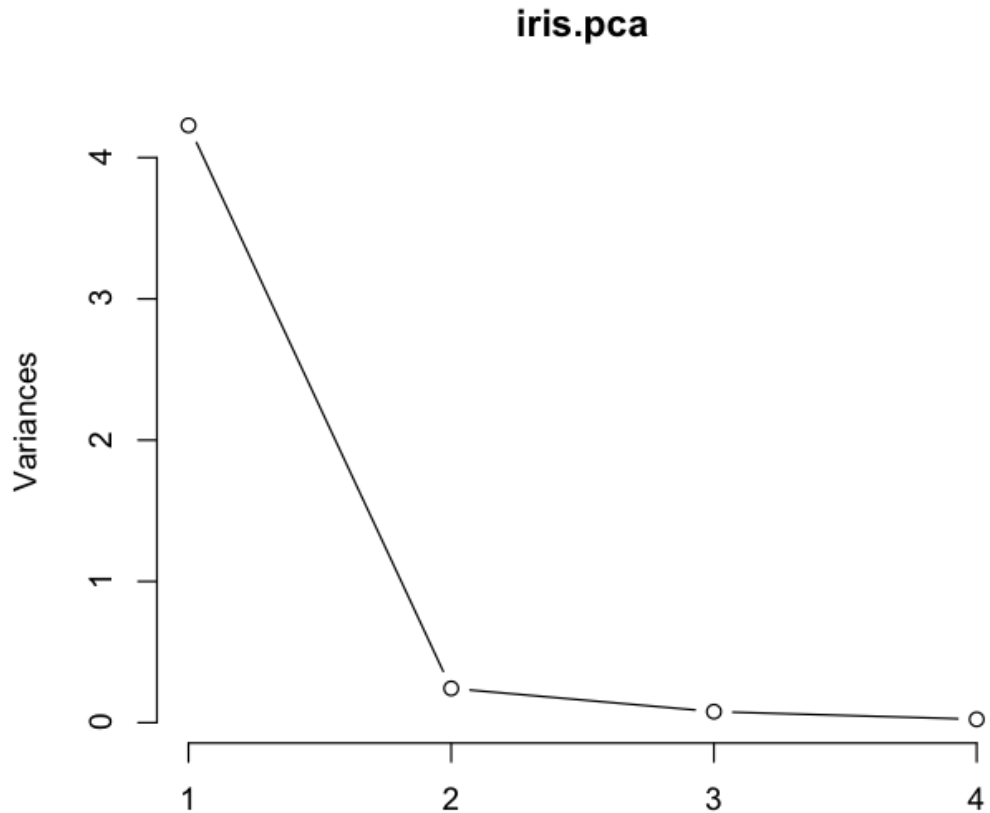
$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}.$$

FIG. 2 Simulated data of (x, y) for camera A. The signal and noise variances σ^2_{signal} and σ^2_{noise} are graphically represented by the two lines subtending the cloud of data. Note that the largest direction of variance does not lie along the basis of the recording (x_A, y_A) but rather along the best-fit line.

source: http://www.snl.salk.edu/~shlens/pca.pdf



source: http://glowingpython.blogspot.it/2011/07/pca-and-image-compression-with-numpy.html



NOTE

Looking at this plot also gives you an idea of how many principal components to keep.

Apply the *elbow test*: keep only those pc's that appear to the left of the elbow in the graph.

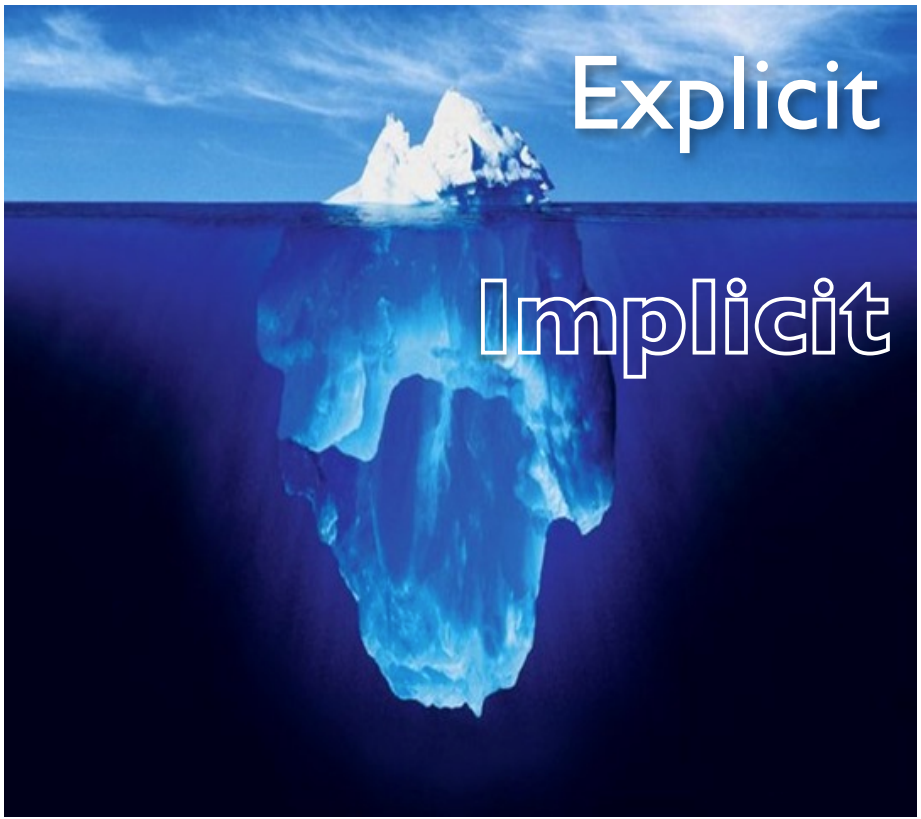
Recommenders need feedback to be useful.

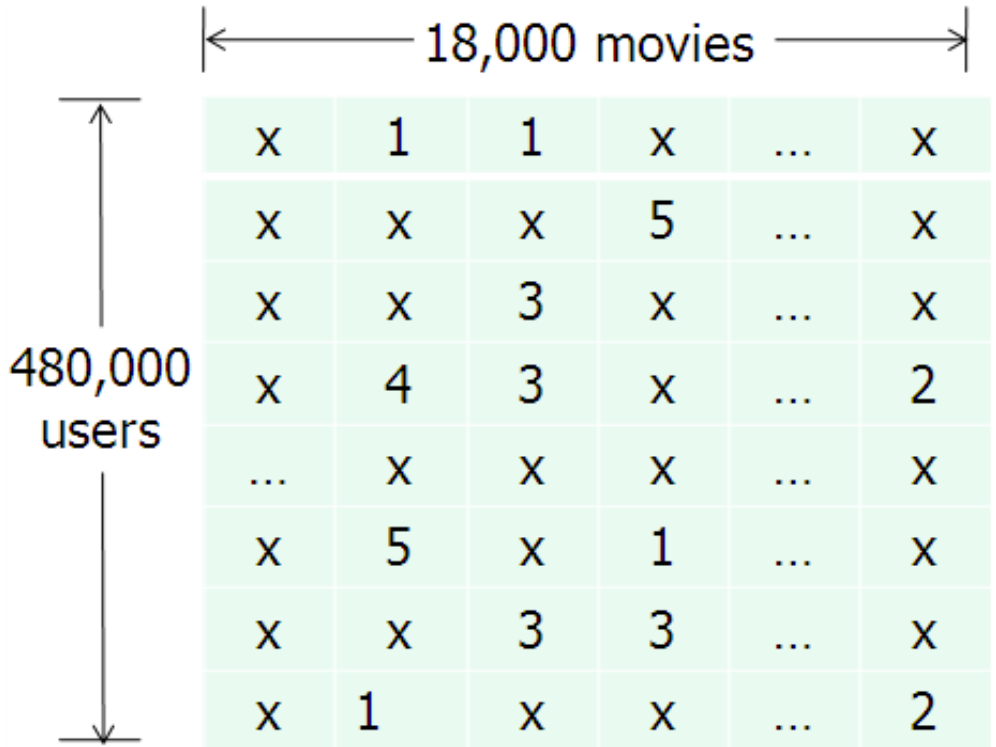
Explicit

- Explicitly given
- Pro-actively acquired
- Expensive to collect

Implicit

- Indirectly given
- Larger quantity
- Latent qualities





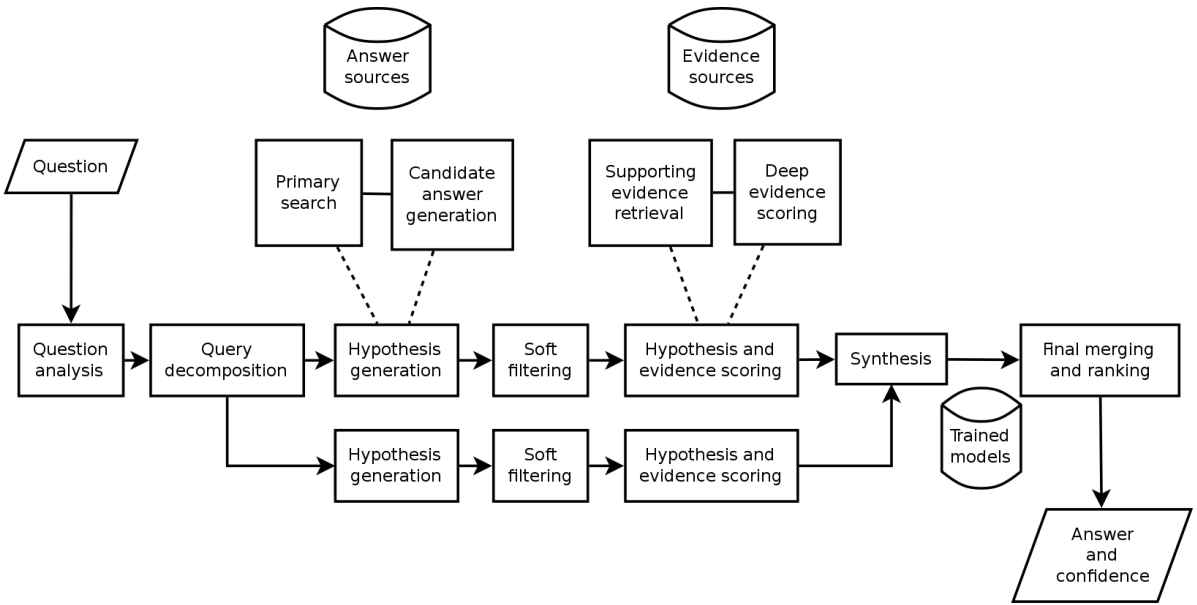
NOTE

This matrix will always be *sparse*!

source: http://www.eecs.berkeley.edu/~zhanghao/main/publications/subfolder/netflix.png

Question Answering

IBM Watson, Wolfram Alpha

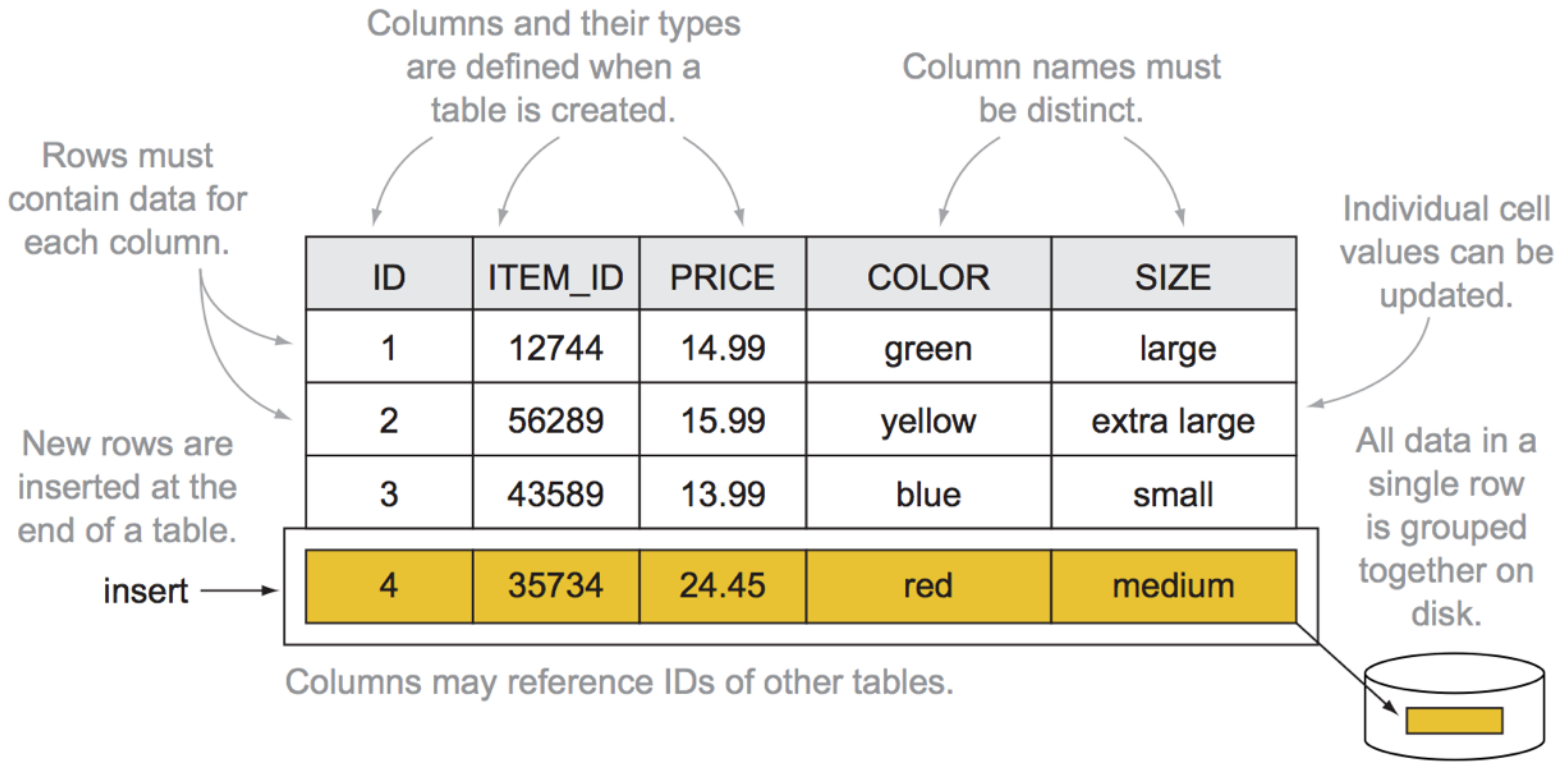


Vs.

Acid	Base

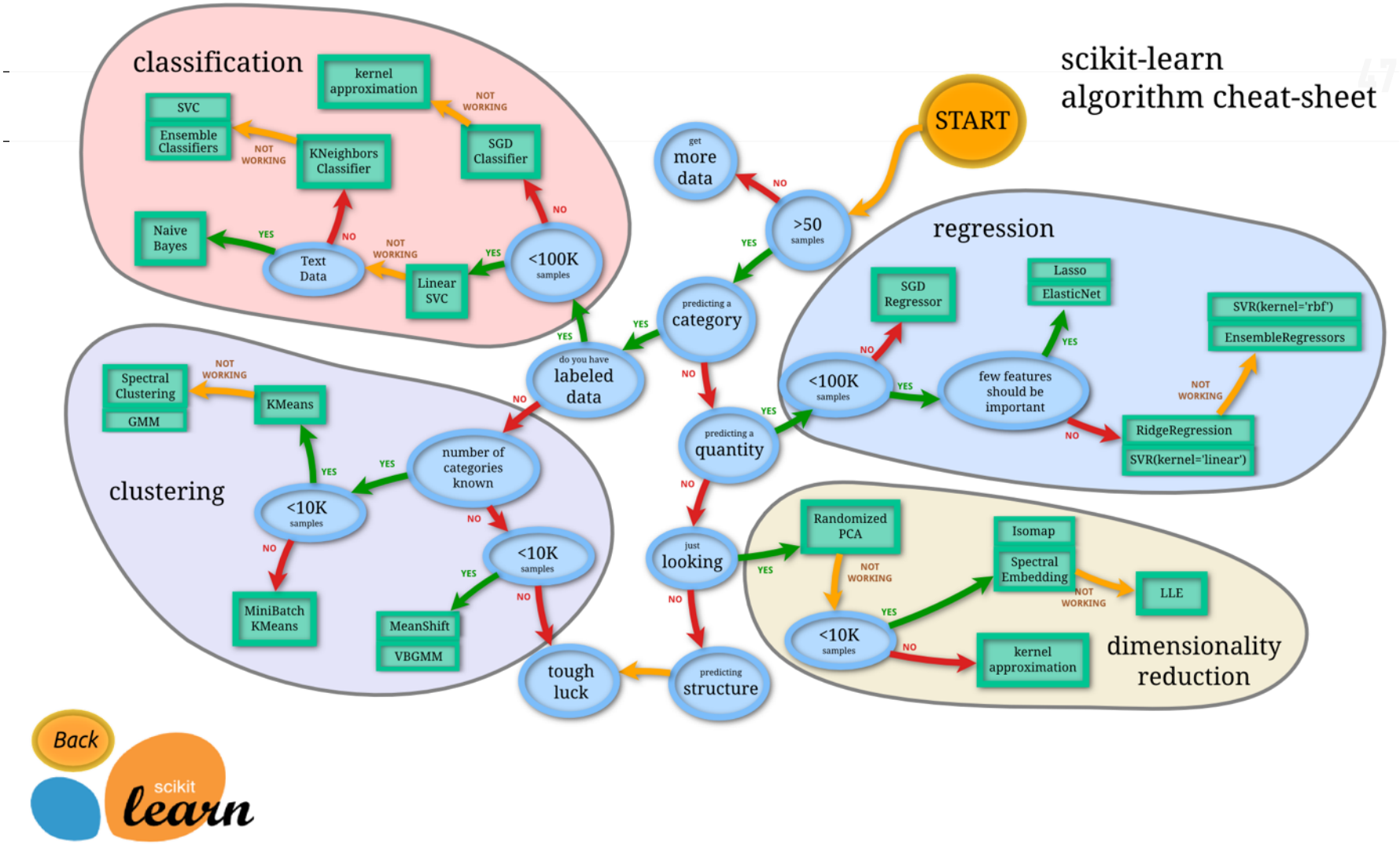
- Get transaction details right
- Block any reports while you are working
- Be pessimistic: anything might go wrong!
- Detailed testing and failure mode analysis
- Lots of locks and unlocks

- Never block a write
- Focus on throughput, not consistency
- Be optimistic: if one service fails it will eventually get caught up
- Some reports may be inconsistent for a while, but don't worry
- Keep things simple and avoid locks


Columns and their types are defined when a table is created.
Column names must be distinct.
Rows must contain data for each column.
Individual cell values can be updated.
ID
ITEM_ID
PRICE
COLOR
SIZE
1
12744
14.99
green
large
2
56289
15.99
yellow
extra large
3
43589
13.99
blue
small
New rows are inserted at the end of a table.
insert
4
35734
24.45
red
medium
All data in a single row is grouped together on disk.
Columns may reference IDs of other tables.

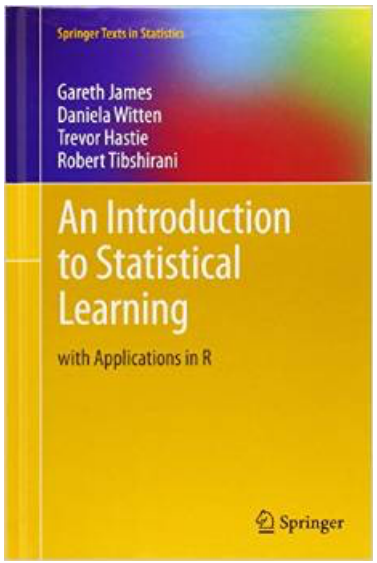
REMEMBER THIS...?

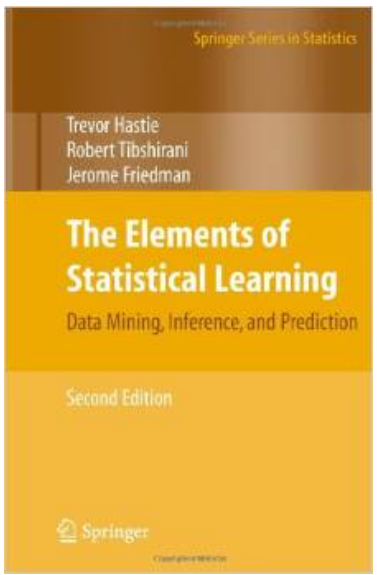
	continuous	***categorical***
supervised	*regression*	*classification*
unsupervised	*dimension reduction*	*clustering*

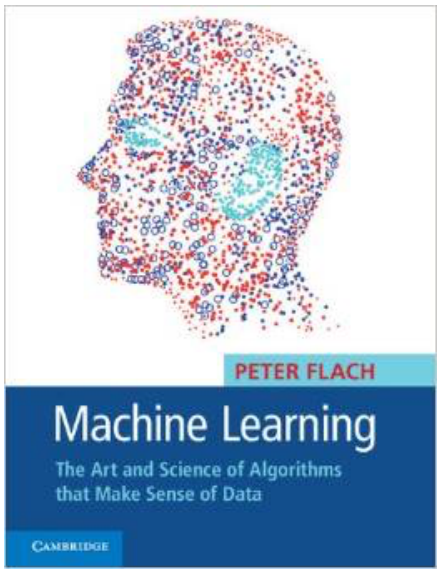

scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
YES
predicting a category
YES
do you have labeled data
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
NO
predicting a quantity
YES
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
NO
<10K samples
YES
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
just looking
YES
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
Back
scikit learn

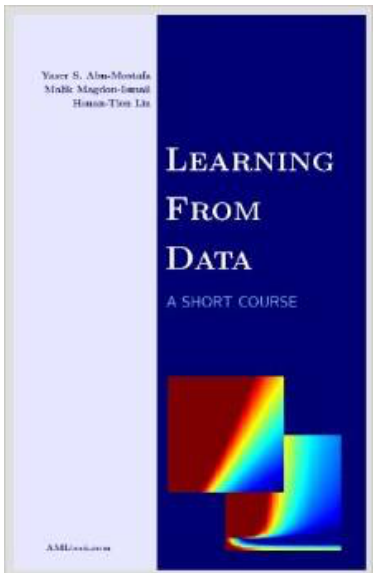
II. WHERE TO GO NEXT?

WHAT TO DO NEXT: READING

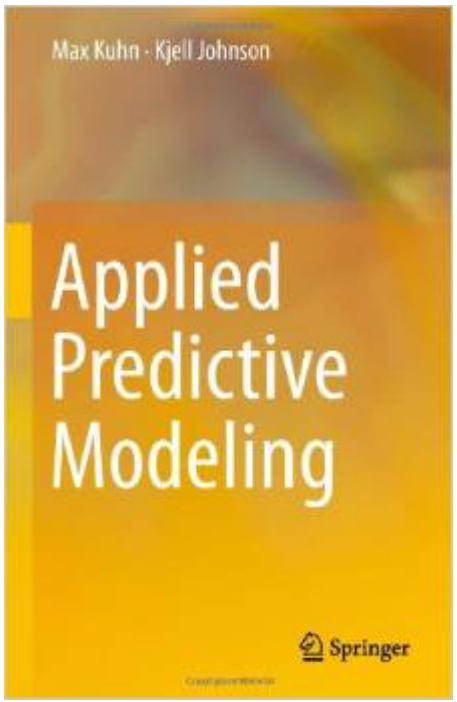


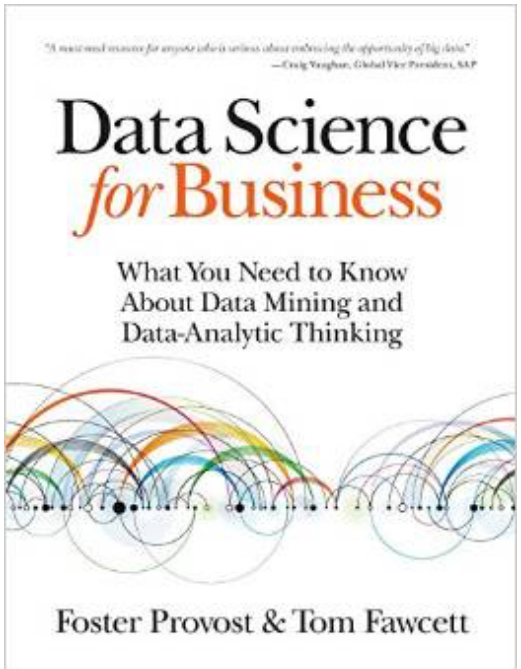


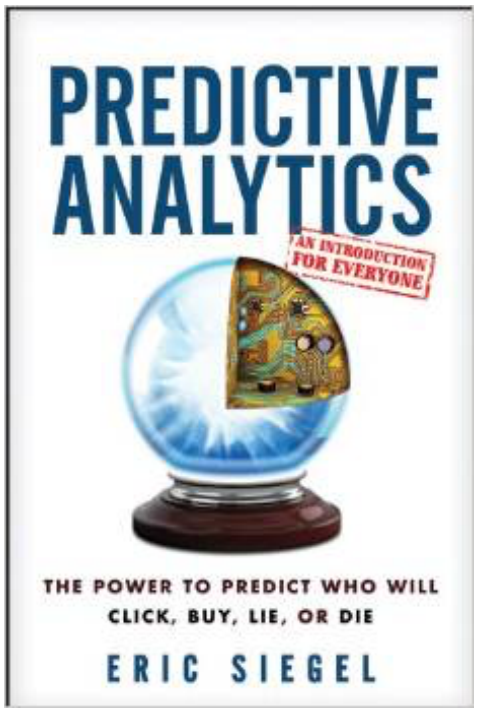




WHAT TO DO NEXT: READING

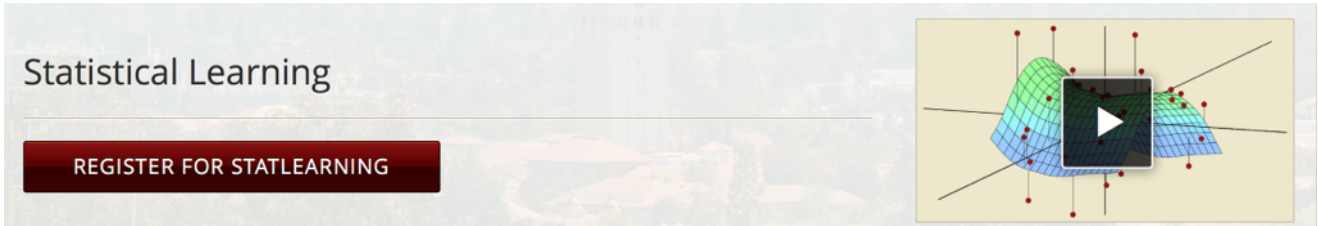







Hadoop
O'REILLY
Programming
Pig
Alan
Programming
Hive
Machine
Learning
for Hackers
O'REILLY
Drew Conway &
John Myles White
Designing
Data
Visualizations
O'REILLY
Noah Iliinsky & Julie Steele
Visualizing
Data
O'REILLY
Ben Fry
Data Analysis
with Open Source Tools
O'REILLY
Philipp K. Janert
Python for
Data Analys
O'REILLY
Wes McKinney
R
O'REILLY
Richard Cotton
Design
Patterns
O'REILLY
Doing Data
Science
STRAIGHT TALK FROM THE FRONTLINE
O'REILLY
Rachel Schutt & Cathy O'Neil

WHAT TO DO NEXT: ONLINE COURSES



Stanford University

Machine Learning

Ended 3 years ago

Course Record

Stanford University

Mining Massive Datasets

Ended 2 months ago

Course Record

WHAT TO DO NEXT: ONLINE COURSES

Stanford University

Natural Language Processing

Ended 3 years ago

Course Record

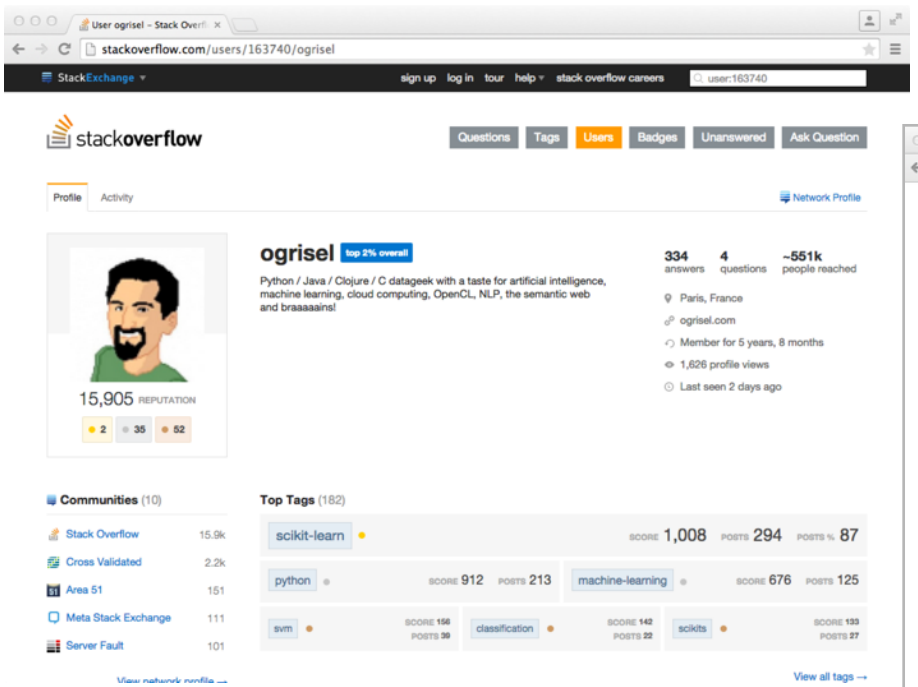
University of Michigan

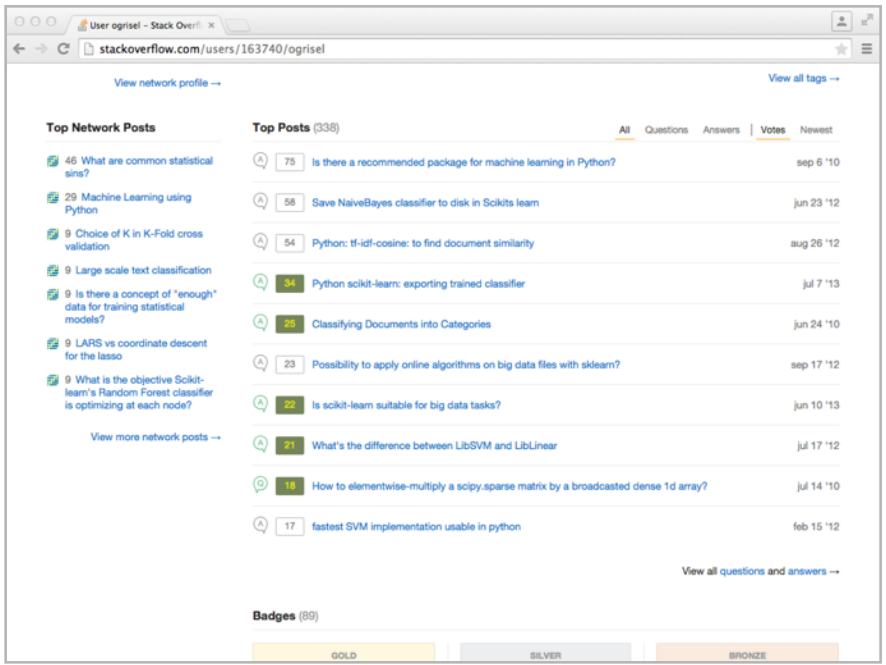
Social Network Analysis

Ended a year ago

Course Record

WHAT TO DO NEXT: STACKOVERFLOW PRO TIP





http://stackoverflow.com/users/163740/ogrisel

- *Python: use it every day / week!*
- *Scala (if you get bored with Python; Spark has PySpark)*
- *Visualization (D3 and js)*
- *Algorithms: Artificial Neural Networks / Deep Learning*
- *"Big Data:"*
  - *MapReduce / Hadoop*
  - *Spark*

kaggle

[topcoder]™

DataKind

- *Online :*
  - *http://www.reddit.com/r/machinelearning*
  - *http://www.r-bloggers.com*
  - *http://dataelixir.com*
  - *http://www.datatau.com*
  - *http://pydata.org*
  - *http://www.technologyreview.com*

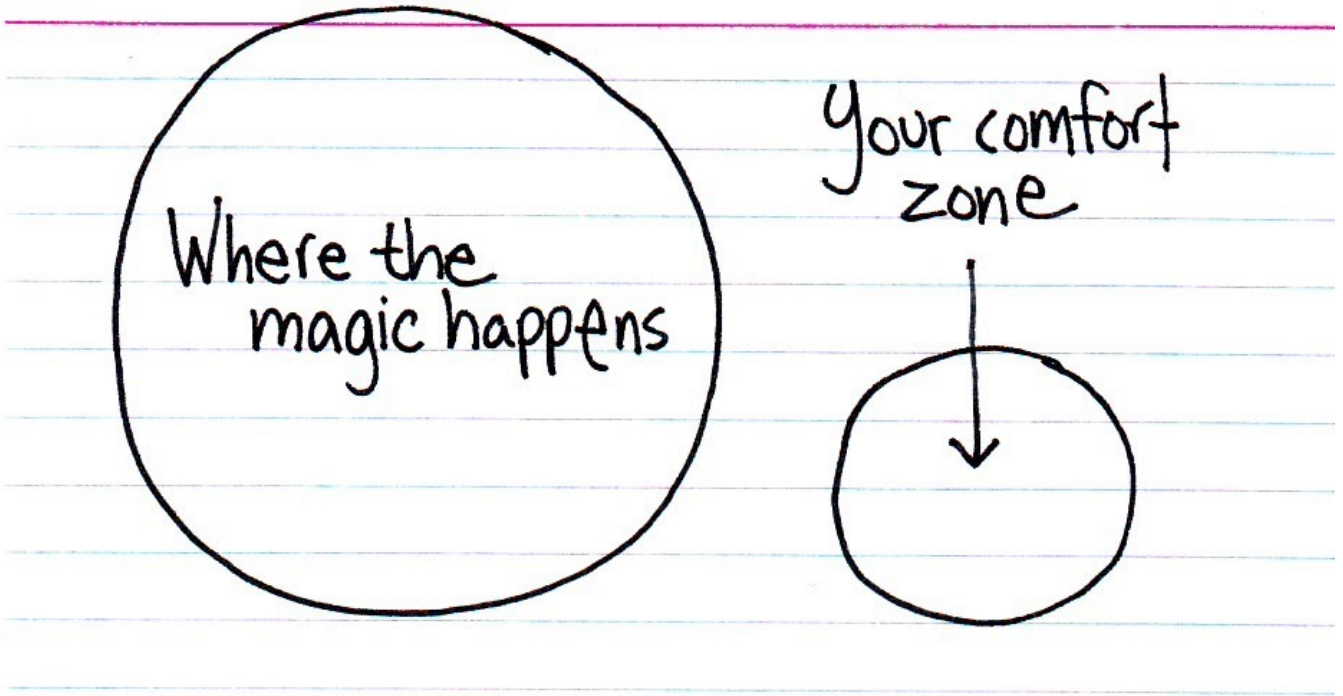
- *LinkedIn groups on ML and DS*

Each Other!!!

- *Class Google group / mailing list*
- *LinkedIn*
- *Study groups (Prior DAT cohorts still have dinner and share job opportunities)*

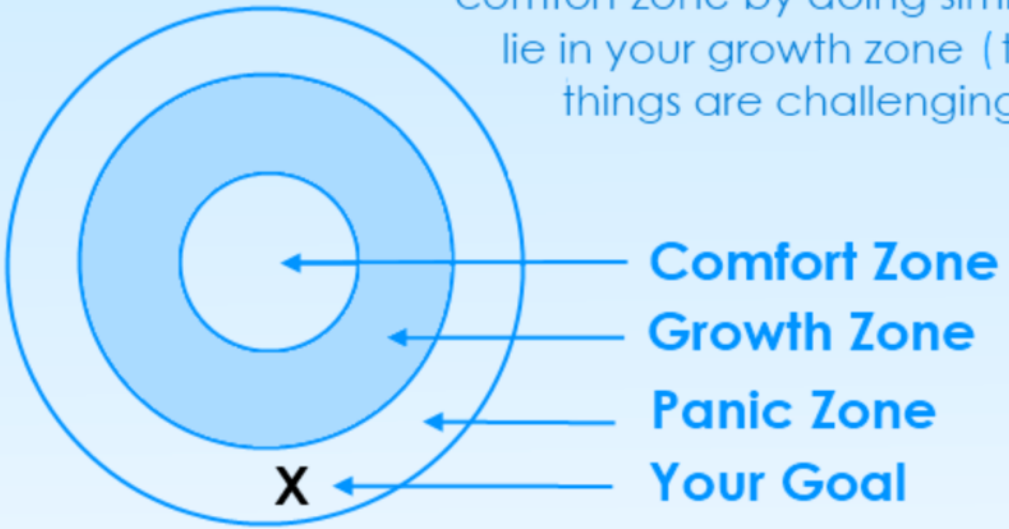
CONGRATULATIONS...

...for getting out of your comfort zone!



How to Grow Your Comfort Zone

Any goal or challenge may fall into one of three zones - your comfort zone, growth zone, or panic zone. If your goal is currently in your panic zone, i.e. it would be too scary to do now, you will need to grow your comfort zone by doing similar challenges that lie in your growth zone (the zone in which things are challenging or scary, but do-able).



How to Grow Your Comfort Zone

As you pursue challenges in your growth zone, those challenges become easier and your comfort zone expands.

Eventually, challenges that were previously in your panic zone begin to fall into your growth zone, and ultimately within your comfort zone.