Engineering Applications of Machine Learning and Data Analytics

Gregory Ditzler

Dept. of Electrical & Computer Engineering
ditzler@email.arizona.edu

Lecture Set Overview

- Course Outline and Expectations
- Definitions and the 30k ft. Perspective
- Applications
- Reading: Chapters 1 & 2 (some of 19 too)

Overview of the Course

“I didn’t have time to write a short letter, so I wrote a long one instead”

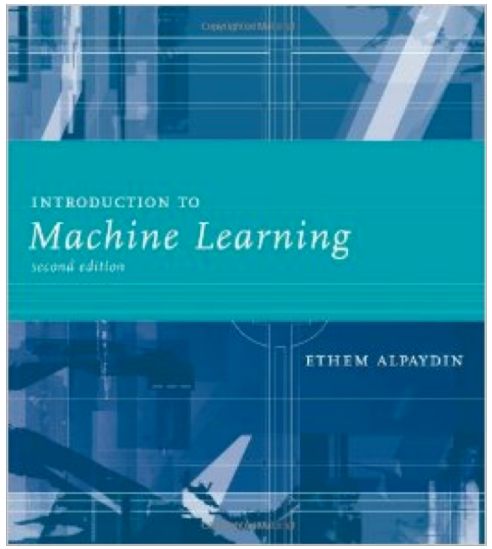
Mark Twain

Topics Covered

- Basics of machine learning
- Supervised and unsupervised learning
- Optimization
- Artificial Neural Networks
  - *Deep learning* and connections to AI
- Support vector machines and empirical risk minimization
- Decision trees and rule-based learning models
- Ensemble learning theory
- Online learning and the multi-armed bandit problem
- Big data
- Applications

Course Textbook

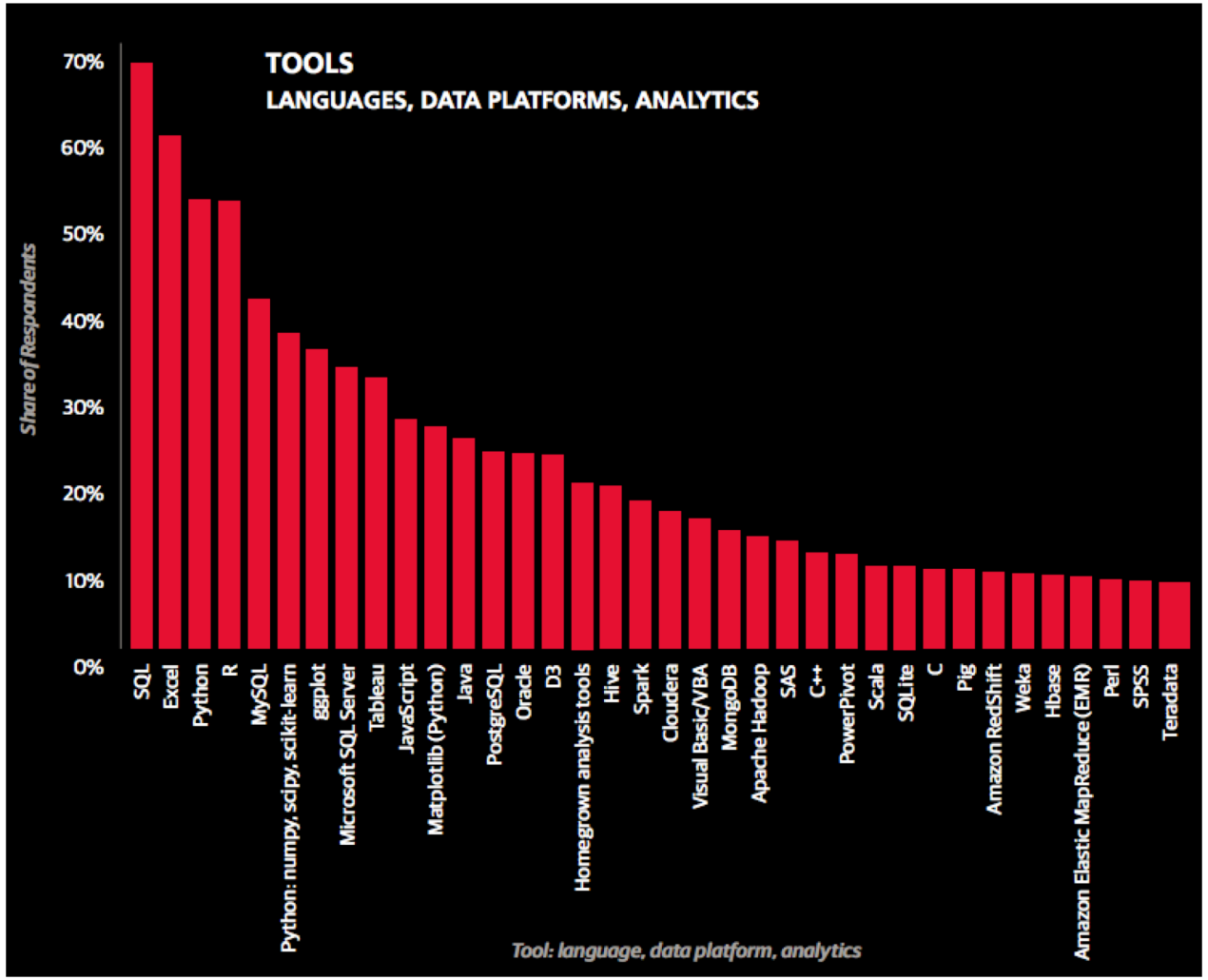
- Required Text
  - “Introduction to Machine Learning,” E. Alpaydin
- Reference
  - “Machine Learning: A Probabilistic Perspective,” K. Murphy
  - “Elements of Statistical Learning Theory,” T. Hastie et al.
  - “Pattern Classification,” R. Duda et al.
  - “Pattern Recognition and Machine Learning,” C. Bishop



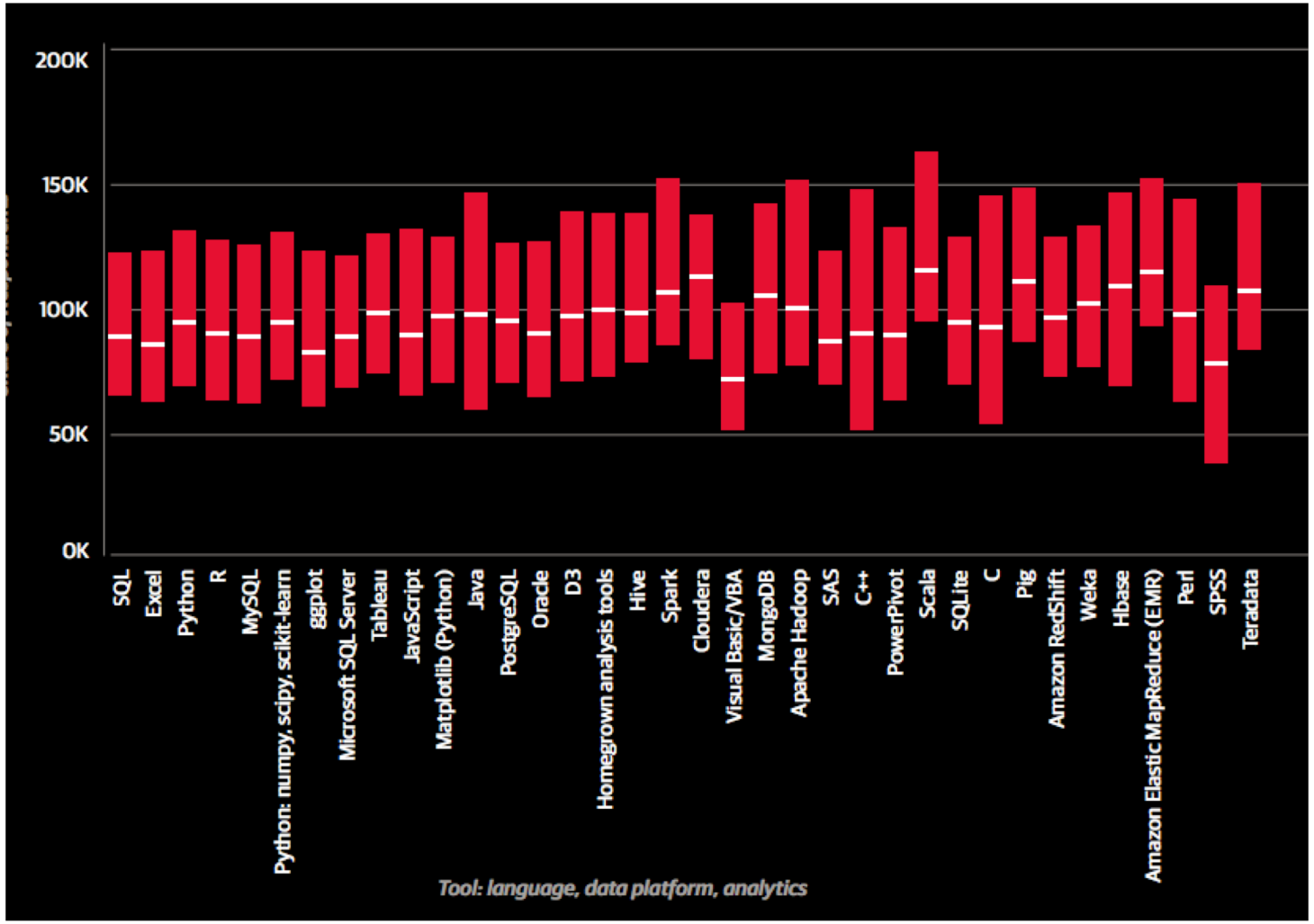
Grading Rubric

- Homework (35%)
  - Approximately five assignments (theory + code)
  - Code must be submitted
- Midterm Exams (35%)
  - Two exams
- Final Project (30%)
  - Must be a (small) research project that is ideally aligned with your research
  - Rule of thumb: quality of a conference paper
  - Groups of no more than two are allowed

Software for Homework and Projects



Software for Homework and Projects



Job Market, you ask?

Tech Giants Are Paying Huge Salaries for Scarce A.I. Talent

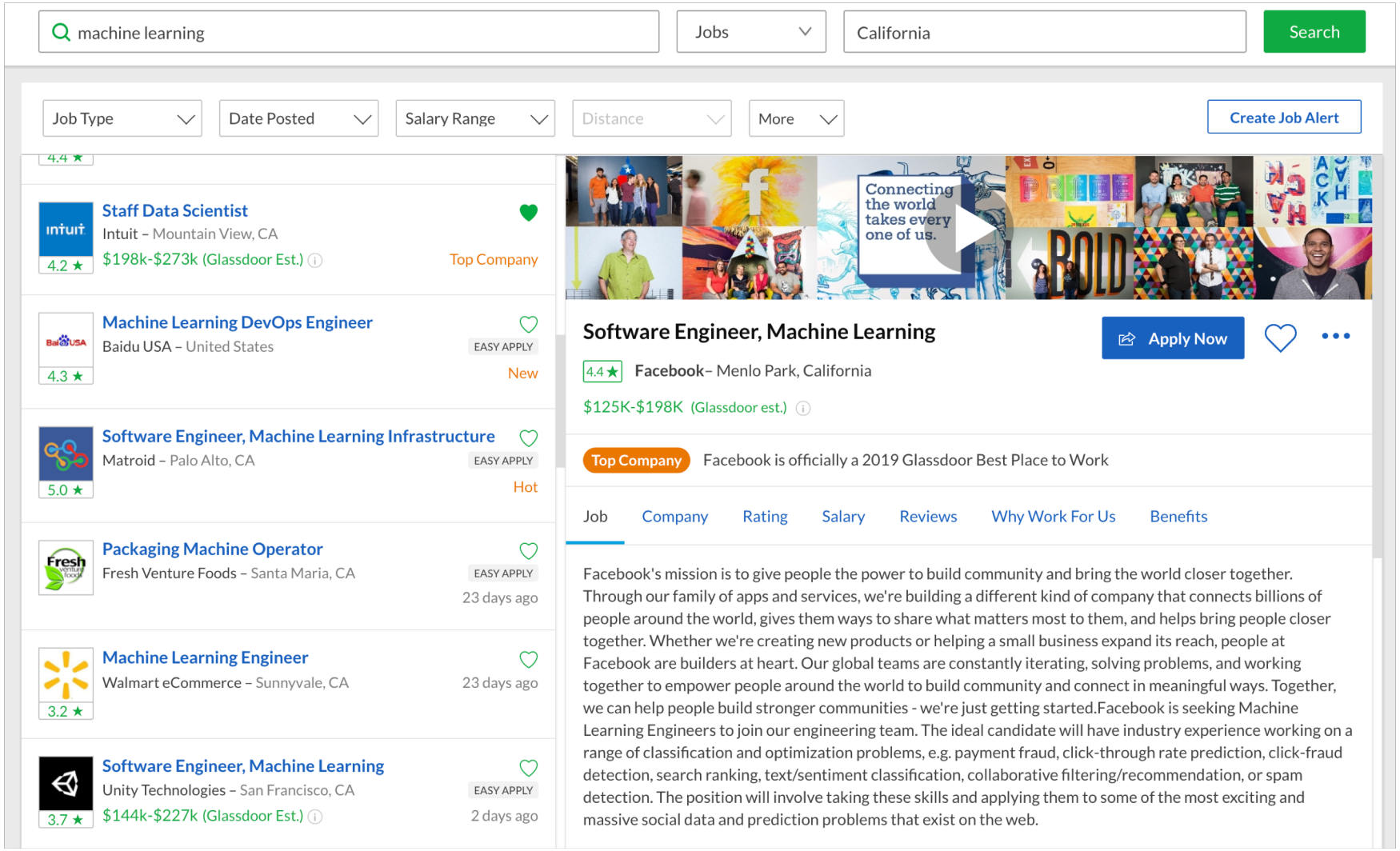
Nearly all big tech companies have an artificial intelligence project, and they are willing to pay experts millions of dollars to help get it done.

查看简体中文版 | 查看繁體中文版

By CADE METZ OCT. 22, 2017

https://www.nytimes.com/2017/10/22/technology/artificial-intelligence-experts-salaries.html

What does Glassdoor have to say?



Software for Homework and Projects

- Python (https://www.continuum.io)
  - Scikit Learn (http://scikit-learn.org/)
  - Tensorflow (https://www.tensorflow.org/)

Introduction to Machine Learning

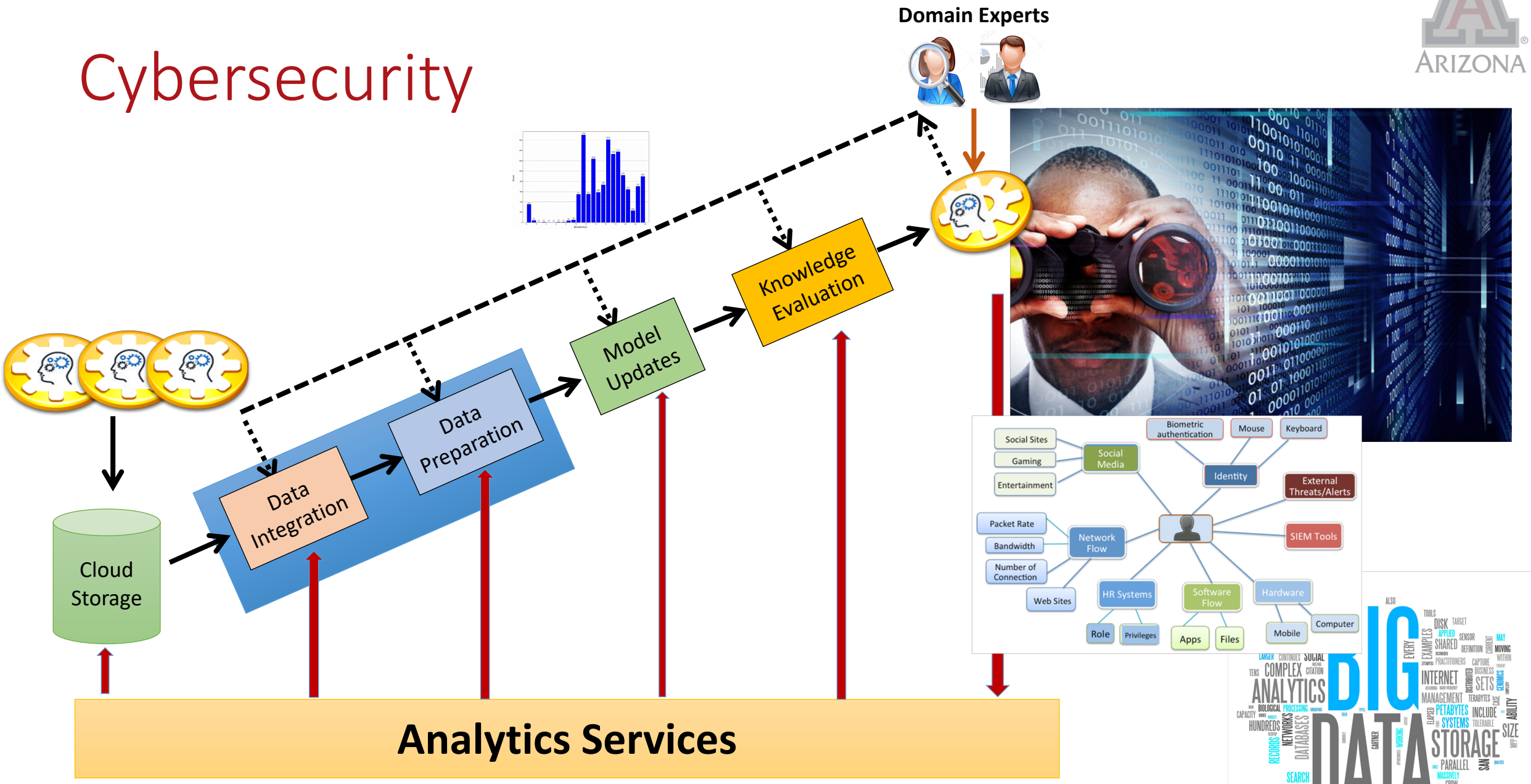
“What is essential is invisible to the eye”
Antoine de Saint Exupery

Machine Learning in a Nutshell

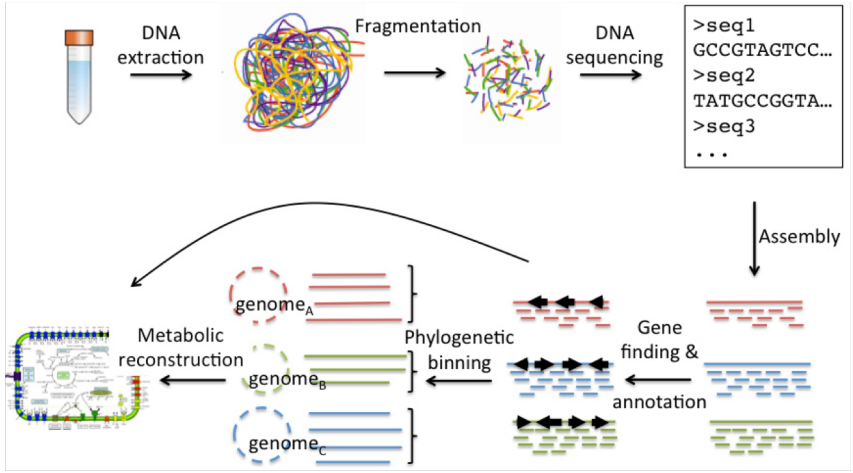
Machine Learning

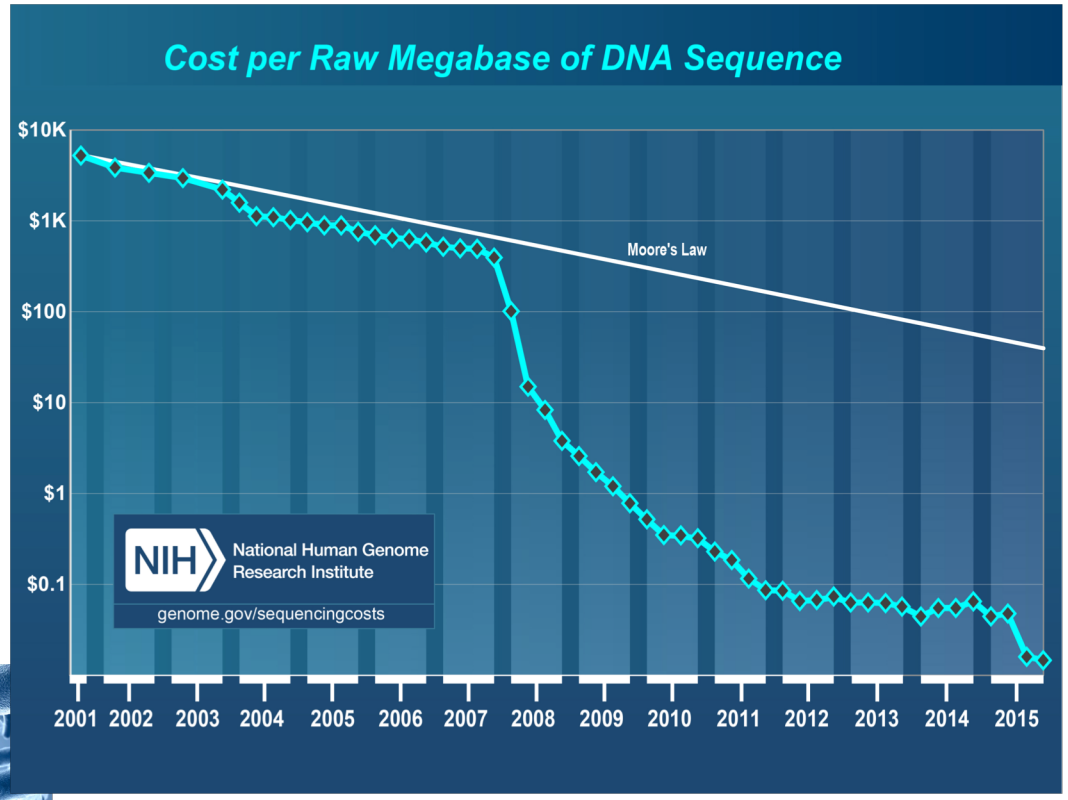
Computer Science	Statistics	Engineering & Optimization	Neuroscience
Artificial Intelligence	Statistical Learning Theory	Computational Intelligence	Computational Neuroscience
Learning Theory	Statistical **Pattern Recognition**	Pattern Recognition	Learning & Memory Models

Cybersecurity



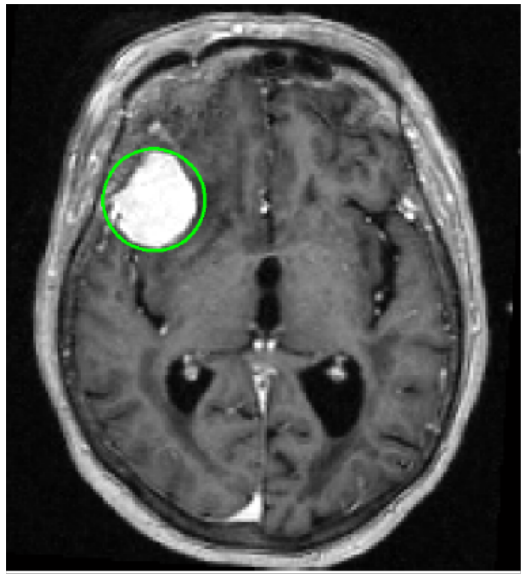
Bioinformatics

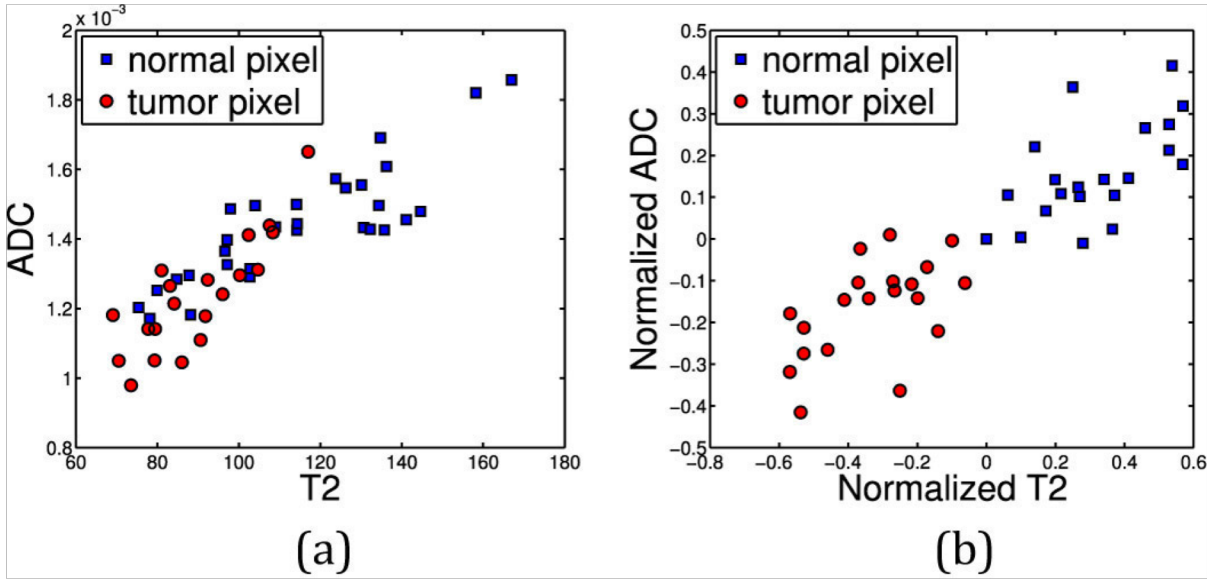


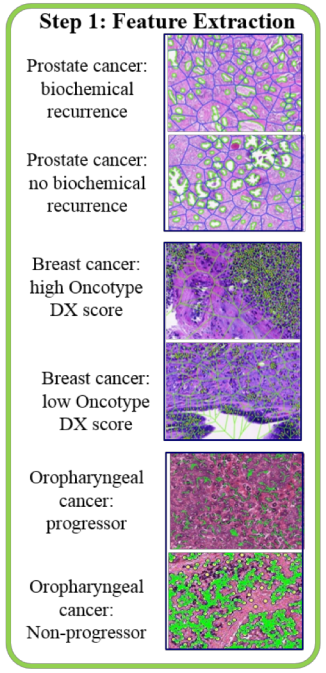


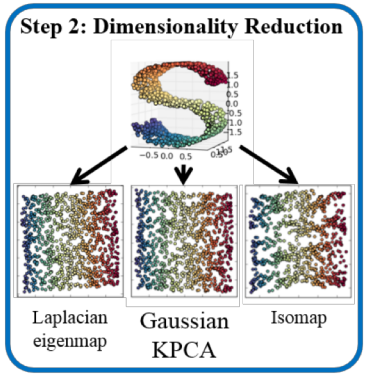
NIH
Big Data to Knowledge(BD2K)

Cancer Screening







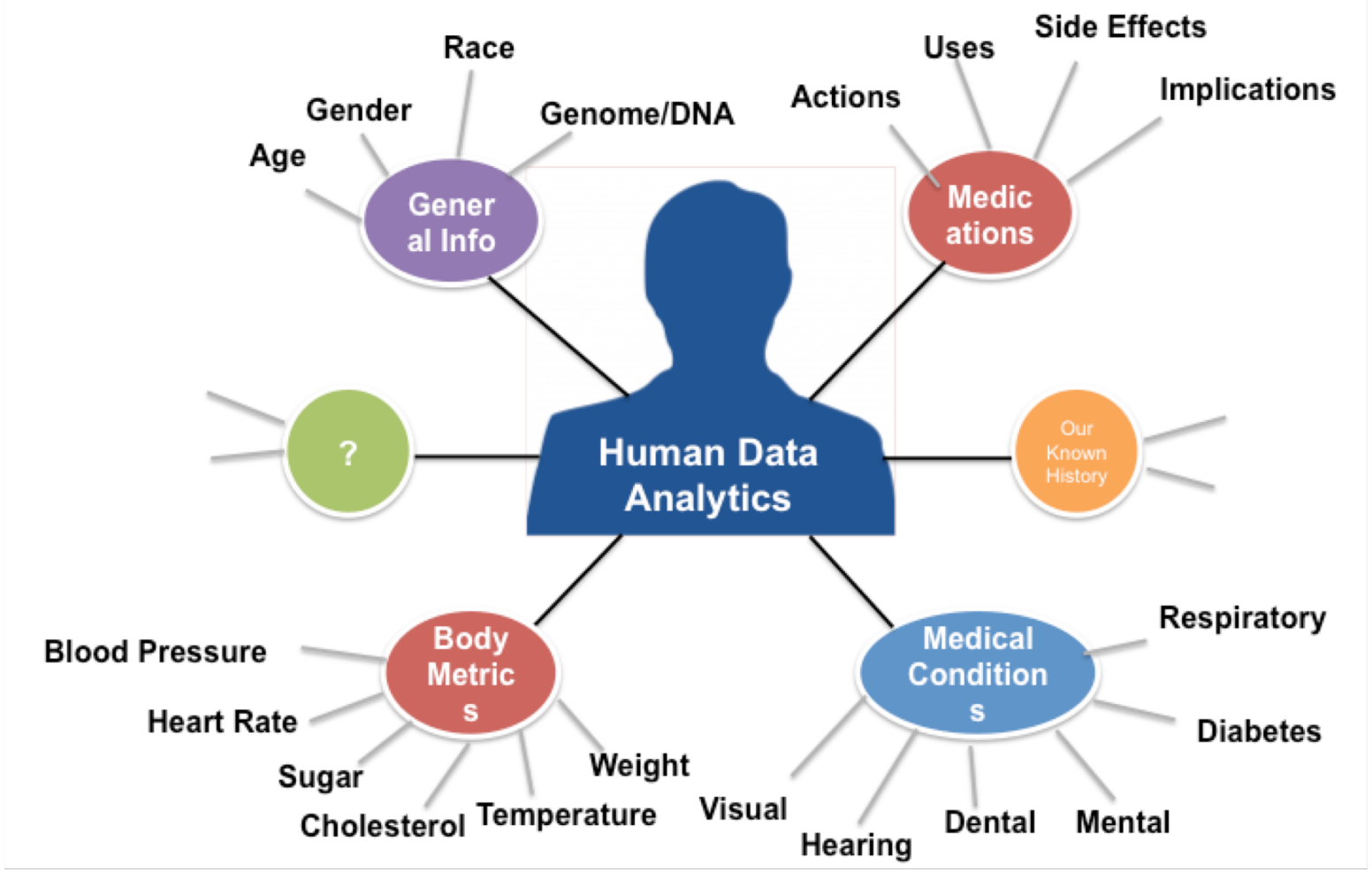


Step 3: Compute VINK

Assess variable contributions to classification on the embedding

$$\pi_j = \sqrt{m \frac{\sum_{i=1}^{h} b_i^2 \mathbf{t}_i^{\mathsf{T}} \mathbf{t}_i \left(\frac{p_{ji}}{\|\mathbf{p}_i\|} \right)^2}{\sum_{i=1}^{h} b_i^2 \mathbf{t}_i^{\mathsf{T}} \mathbf{t}_i}}$$

Healthcare Informatics



Economics

ECONOMICS

Economics Has a Math Problem

70 SEPT 1, 2015 8:00 AM EDT

By Noah Smith

"Two economists who have been pushing for the adoption of machine learning techniques in economics are Susan Athey and Guido Imbens of Stanford University. The two economists explained machine learning techniques to an interested crowd at a recent meeting of the National Bureau of Economic Research. Their overview stated that machine learning techniques emphasized causality less than traditional economic statistical techniques, or what's usually known as econometrics. **In other words, machine learning is more about forecasting than about understanding the effects of policy**."

So is economics going to become another branch of applied math? Will econometrics and data science merge? **Berkeley economist Brad DeLong thinks so. "The work [of economics] will be done," he writes, "by data scientists, computer modelers, and historians of various stripes."** That is almost certainly too extreme a prediction. But the interest in machine learning is just one more **sign that economics may be starting to shed its peculiar fixation on theory and join its cousins in the data-driven future.**

BloombergView

BIG DATA

VOLUME

DATA SIZE

VELOCITY

SPEED OF CHANGE

VARIETY

DIFFERENT FORMS OF DATA SOURCES

VERACITY

UNCERTAINTY OF DATA

Preliminary Material & Terminology

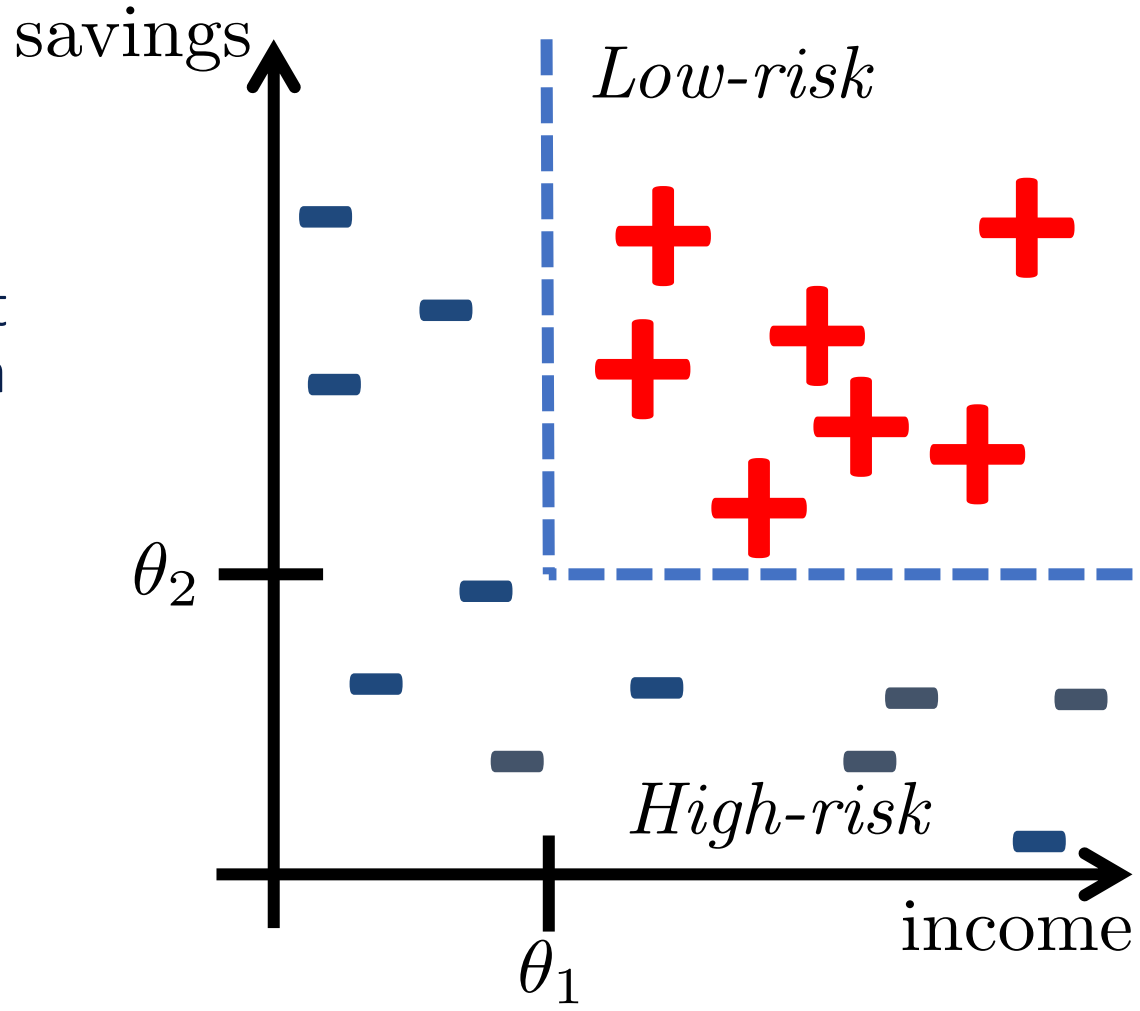
What is Machine Learning

- Informal definition: Automated analysis of – typically large volumes of – data in search of hidden structures / patterns / information
  - Pattern recognition: Classification of objects into (predefined) categories or classes
    - Given data, assign labels (categories) that identify the correct class
    - Identify the input / output relationship (mapping) of an unknown system (system identification)
- Jobs affiliated with the analysis of massive volumes of data are on the rise and there is a need to fill these positions with competent, clever and knowledgeable individuals

Terminology

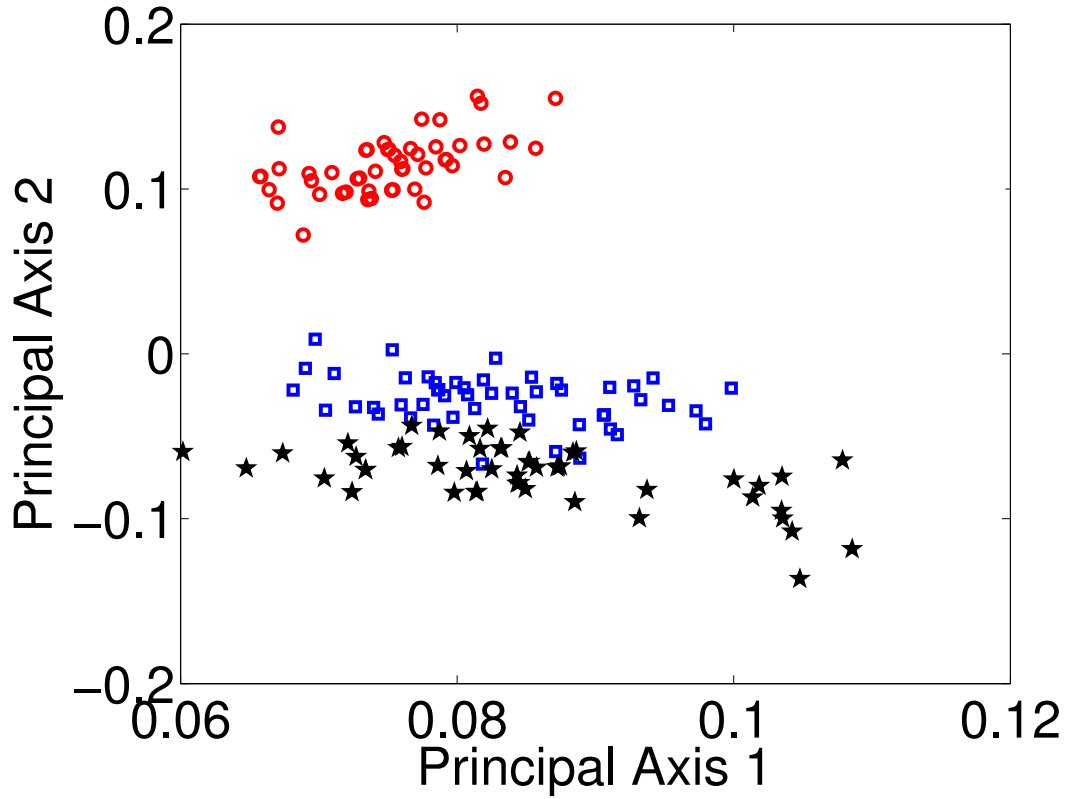
- Classification: Identification (assigning a label) of a particular object to its correct category based on the (features of) data collected from that object.
  - Classify handwritten objects into one of previously fixed set of alpha-numerical characters
  - Loan / credit applications

IF income > θ1 AND savings > θ2 THEN
  low-risk loan
ELSE
  high-risk loan



Terminology cnt'd

- Clustering: Given data of objects obtained from an unknown number and nature of categories, grouping of such data into clusters based on some measure of similarity
  - Data mining: Given large volumes of data obtained from the web pages, group the corresponding web pages into logically meaningful groups (e.g., news articles, shopping sites, medical information, etc.)



Terminology cnt'd

- Supervised learning: Given training data with previously labeled classes, learn the mapping between the data and their correct classes.
  - Associated with "classification," typically involves adaptively changing the parameters of a model (classifier) until the model output fits the data
- Unsupervised learning: Given unlabeled data obtained from unknown number of categories, learn how to group such data into meaningful clusters based on some measure of similarity
  - Typically associated with "clustering" and "density estimation"
- Reinforcement learning: Given a sequence of outputs, learn a policy to obtain the desired output. Typically associated with credit assignment and game playing problems
  - No single good move - game is won, if the sequence of moves are collectively good!

Terminology cnt'd

- Feature: a variable believed to carry discriminating and characterizing information about the objects under consideration; a.k.a. *predictor*, *attribute*, *covariate*
- Feature vector: A collection of *d* features, ordered in some meaningful way into a *d-dimensional* column vector, that represents the signature of the object to be identified.
- Feature space: The *d-dimensional* space in which the feature vectors lie. A *d-dimensional* vector in a *d-dimensional* space constitutes a point in that space.

Terminology cnt'd

- Class: The category to which a given object belongs, typically denoted by ω *or* y
- Pattern: A collection of features of an object under consideration, along with its correct class information. In classification, a pattern is a pair of variables, $\{\vec{\mathbf{x}}_i, \omega_i\}$ (or $\{\vec{\mathbf{x}}_i, y_i\}$) where $\vec{\mathbf{x}}_i$ is the i^{th} feature vector, and ω_i (or y_i) is the corresponding label.
- Instance/Sample/Exemplar: Any given example pattern of an object
- Decision boundary: A boundary in the *d-dimensional* feature space that separates patterns of different classes from each other.

Terminology cnt'd

- Training Data: Data used during training of a classifier for which the correct labels are *a priori* known
- Test / Validation Data: Data not used during training, but rather set aside to estimate the true (generalization) performance of a classifier, for which correct labels are also *a priori* known
- Field Test Data: Unknown data to be classified for which the classifier is ultimately trained. The correct class labels for these data are not known *a priori.*

Terminology cnt'd

- Cost Function: A quantitative measure that represents the cost of making an error. The classifier is trained to minimize this function.
- Classifier: A parametric or nonparametric model which adjusts its parameters or weights to find the correct decision boundaries through a learning algorithm using a training dataset – such that a cost function is minimized.
- Model: A simplified mathematical / statistical construct that mimics (acts like) the underlying physical phenomenon that generated the original data.
- Parametric Model: A probabilistic / statistical model that assumes that the underlying phenomenon follows a specific known probability distribution. The parameters of such a model are the parameters of the distribution.
  - A classifier based on determining the parameters of a distribution is also called a ***generative model*** as the underlying distribution can be generated from the parameters.
  - Examples: Bayes classifier
- Nonparametric model: A model that does not assume a specific distribution, and that typically follows an optimization algorithm to minimize error.
  - A classifier based on using a nonparametric approach is also called a ***discriminative model***, as the decision is then based on a ***discriminant*** (or discriminant function).
  - Examples: Neural networks, decision trees, support vector machines.

Terminology cnt'd

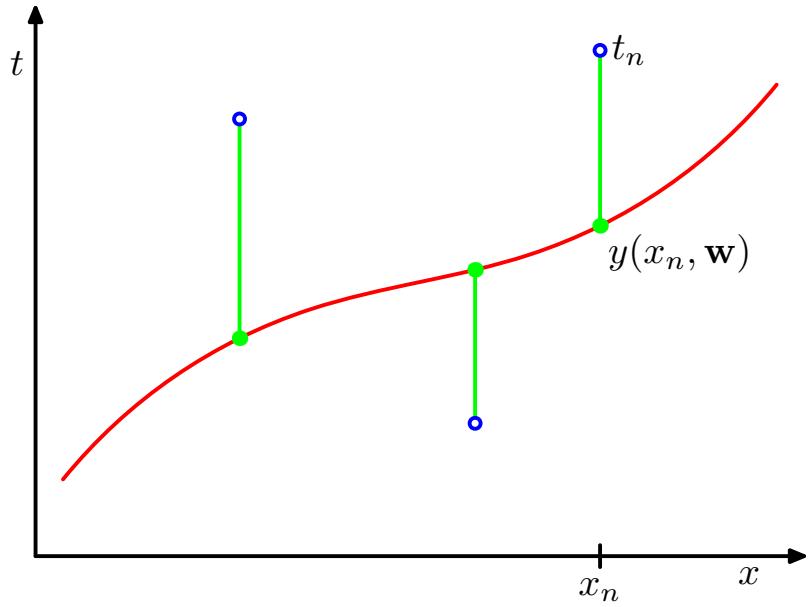
- Error: Incorrect labeling of the data by the classifier
- Cost of error: Cost of making a decision, in particular an incorrect one – not all errors are equally costly!
- Training Performance: The ability / performance of the classifier in correctly identifying the classes of the training data, which it has already seen. It may not be a good indicator of the generalization performance.
- Generalization (Test Performance): The ability / performance of the classifier in identifying the classes of previously unseen patterns.
- Confusion Matrix: The matrix obtained from test performance of the classifier that shows how many instances of each class are classified into different classes.

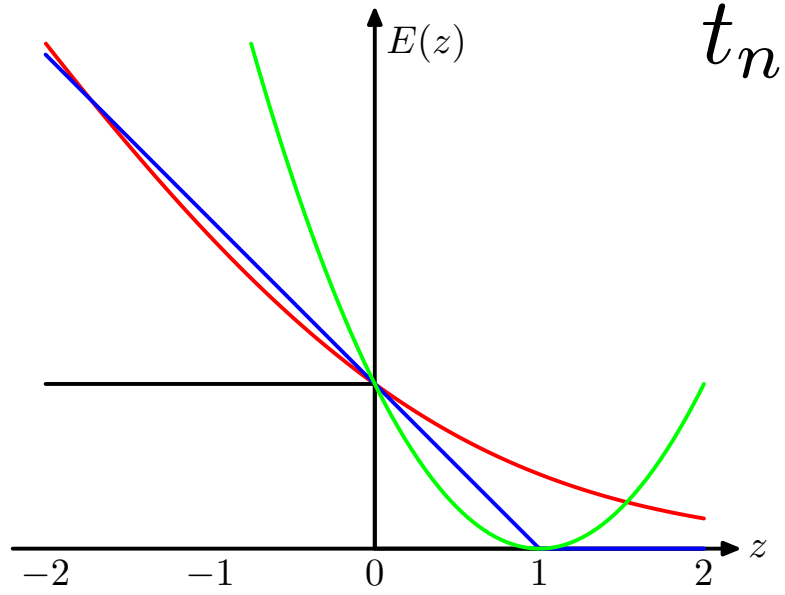
Measuring Error

$E(\mathbf{w}) \rightarrow$ Error function

$y(x_n, \mathbf{w}) \rightarrow$ Mapping function

$t_n \rightarrow$ Target





$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n \right)^2$$

1-0 Error

Hinge Loss

Squared Loss

Log Loss

} Convex

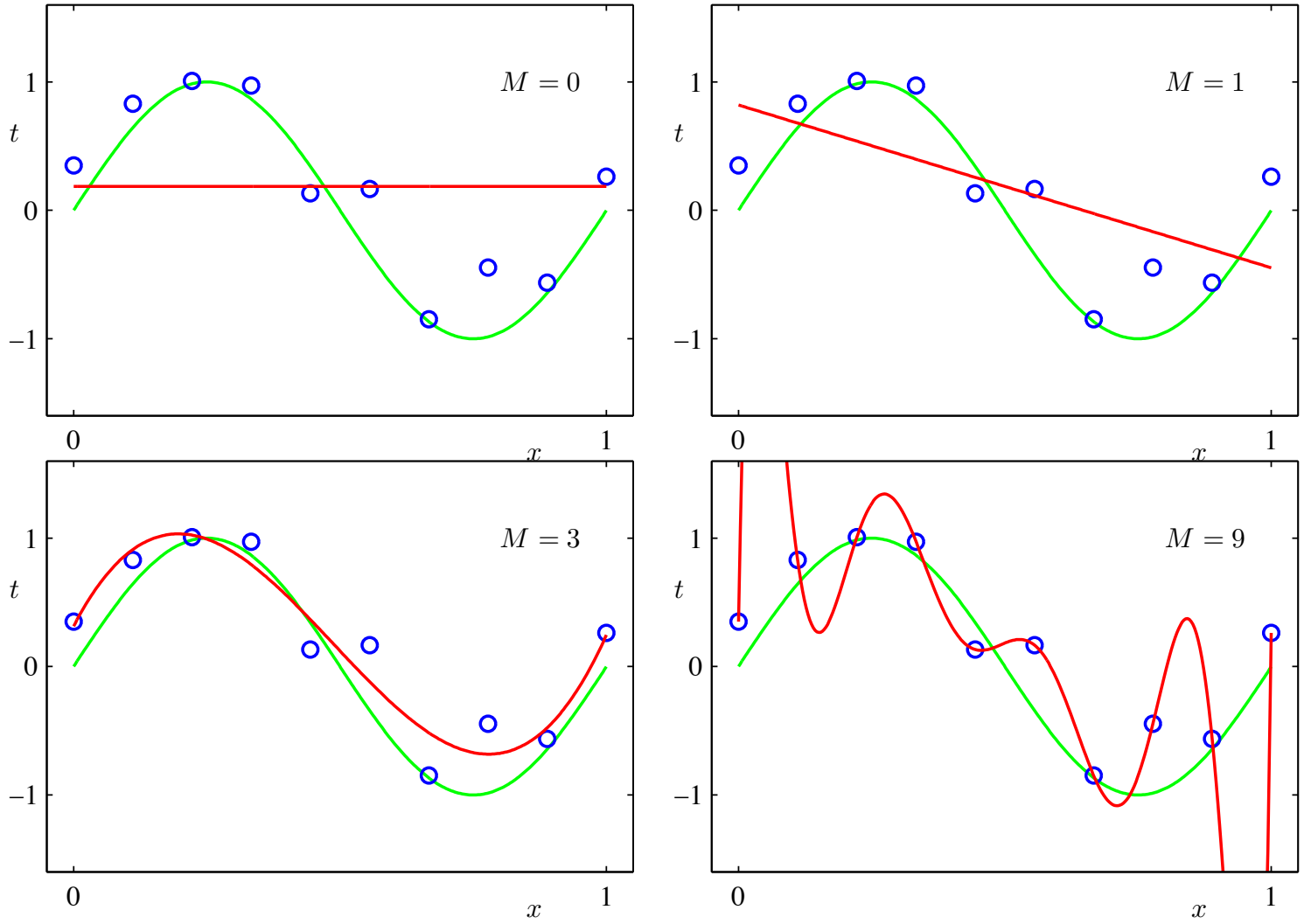
Example: Fitting a Polynomial

$$y(x_n, \mathbf{w}) = \sum_{j=0}^{M} w_j \, x^j$$

- Consider y(x, w) taking the form of a polynomial function and the squared error?
  - Is zero error a good thing?
  - How should we choose the polynomial M?

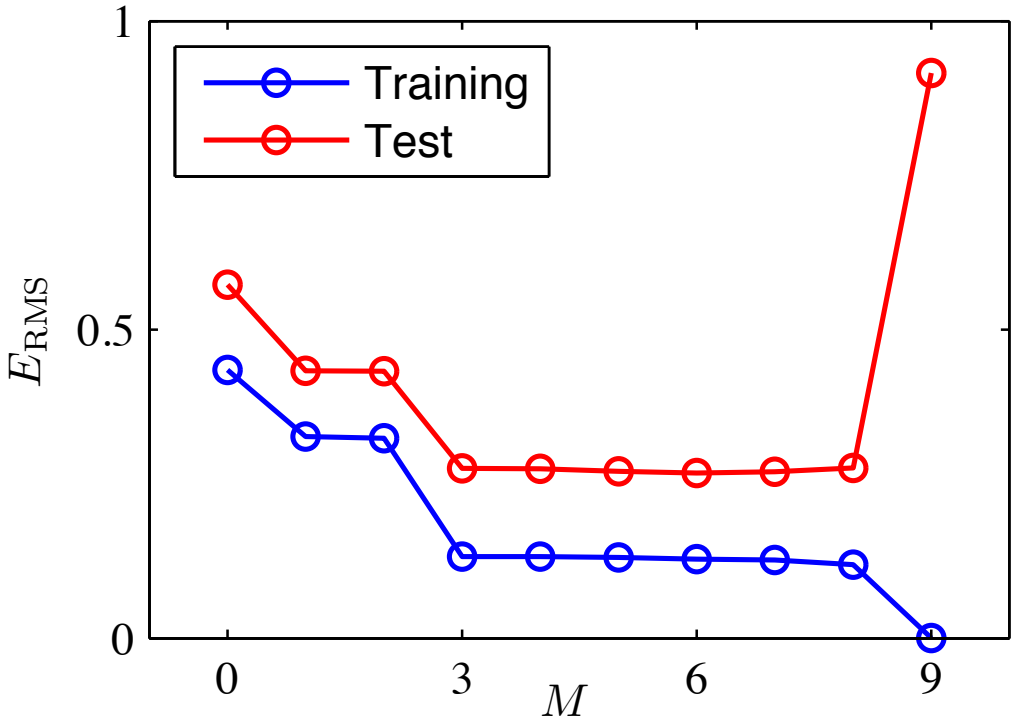
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n \right)^2$$

Choosing M and Overfitting



Overfitting

- Training error goes to zero, but the testing error is quite large
- Measurements of error on the training data can be deceiving! This is why we use validation data to help us detect overfitting

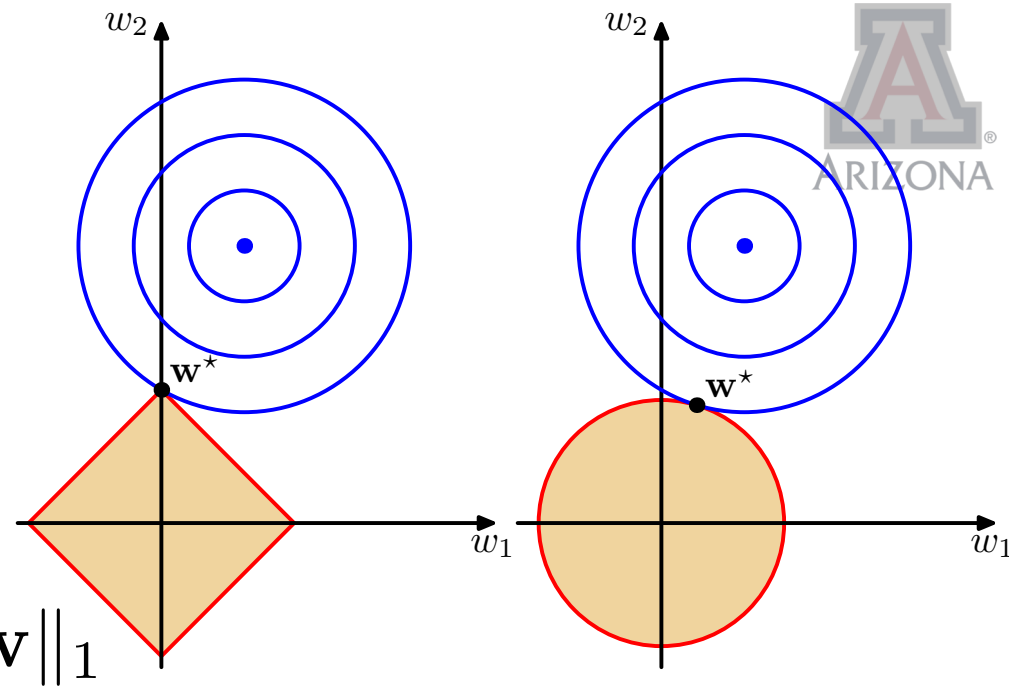


$$y(x_n, \mathbf{w}) = \sum_{j=0}^{M} w_j \, x^j$$

Keeping Overfitting Under Control

- Many models and prediction algorithms suffer from overfitting; however, we can try to avoid overfitting by taking certain precautions.
- Using a *Bayesian approach* can avoid overfitting even when the number of parameters exceeds the number of data points for training.
- *Regularization* is the most commonly used approach to control overfitting. Essentially, we can add a penalty to the error function that discourages the solution vector to take on large values. Yup, its that simple! (for the most part).

Regularization



- L1

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\mathbf{w}) - t_n\right)^2 + \lambda_1\|\mathbf{w}\|_1$$

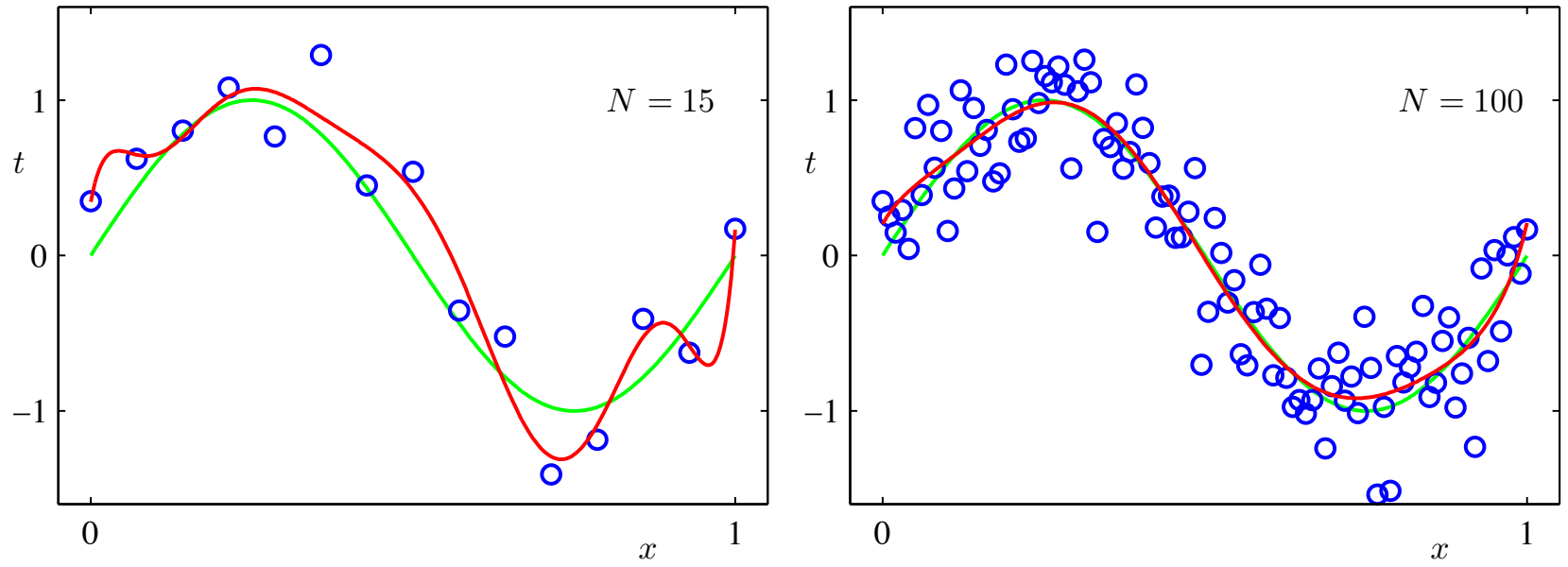
- L2

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\mathbf{w}) - t_n\right)^2 + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$$

- L1+L2

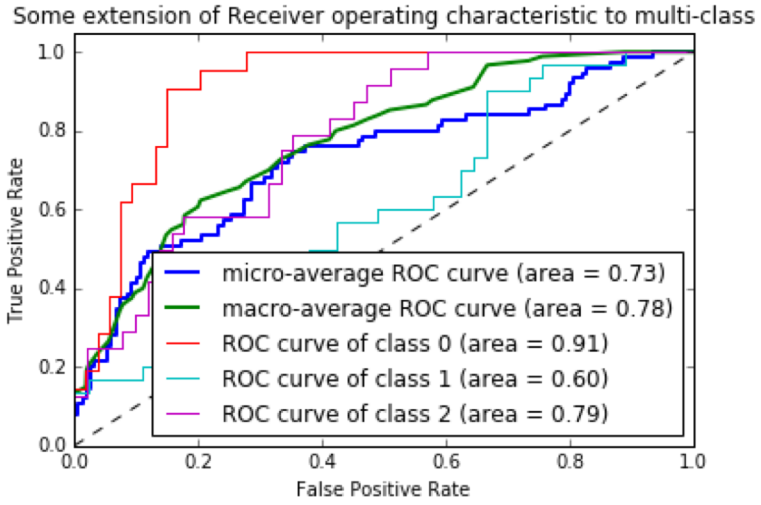
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n,\mathbf{w}) - t_n\right)^2 + \lambda_1\|\mathbf{w}\|_1 + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$$

How Much Data Do I Need?



The green line is the target function, the red function is the result of a 9th order polynomial minimizing ERMS, and the blue points are observations sampled from the target function.

Figures of Merit



Receiver operating characteristic (ROC) curves are used to show the trade off between true positive and false positive rates.

		True	
		+	-
Predicted	+	TP	FP
	-	FN	TN

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{precision} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{f-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

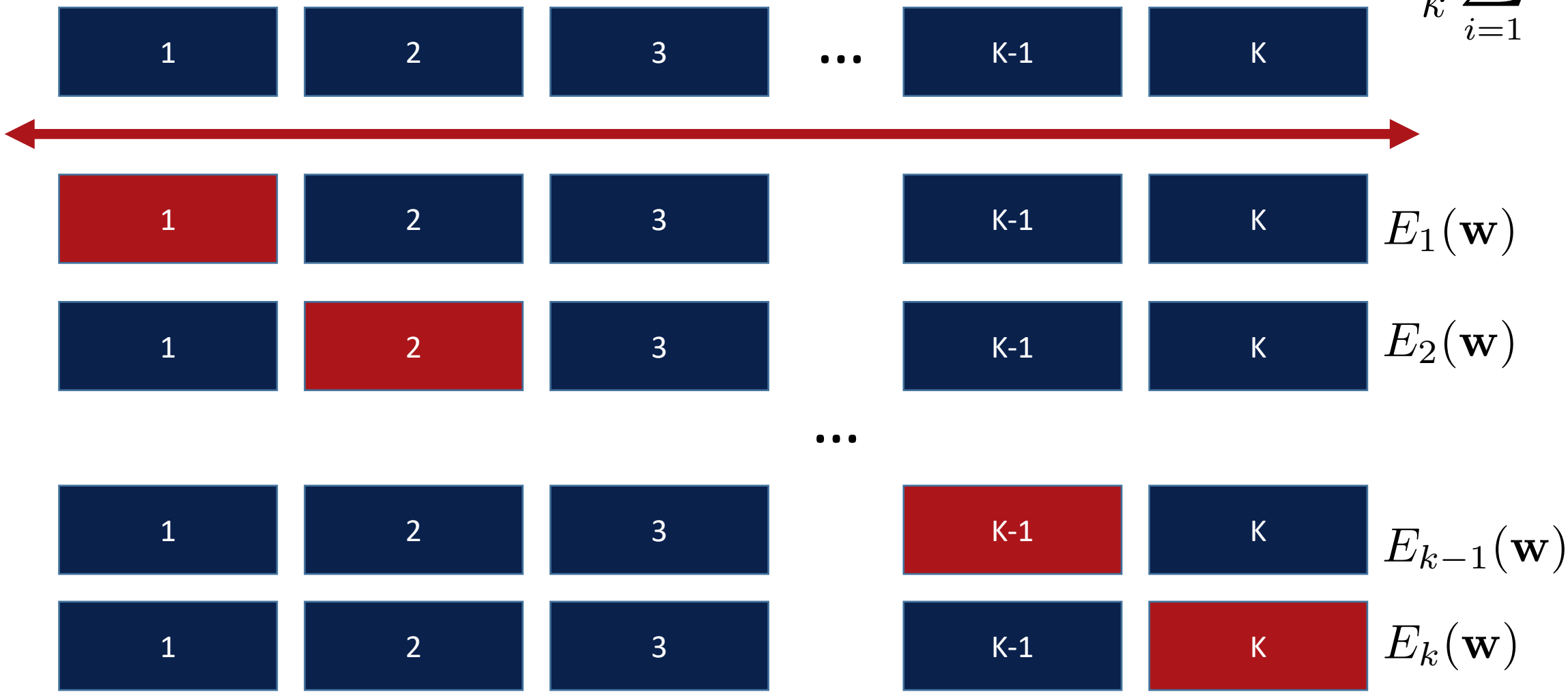
K-Fold Cross Validation

- If we have one training and one testing data set, we can measure the error. Is one measurement enough?
  - Three certainties in life: death, taxes, and noise in your data
- More data sets would give us the opportunity to measure the error several times. An average of these measurements could a better estimate than a single measurement

K-Fold Cross Validation

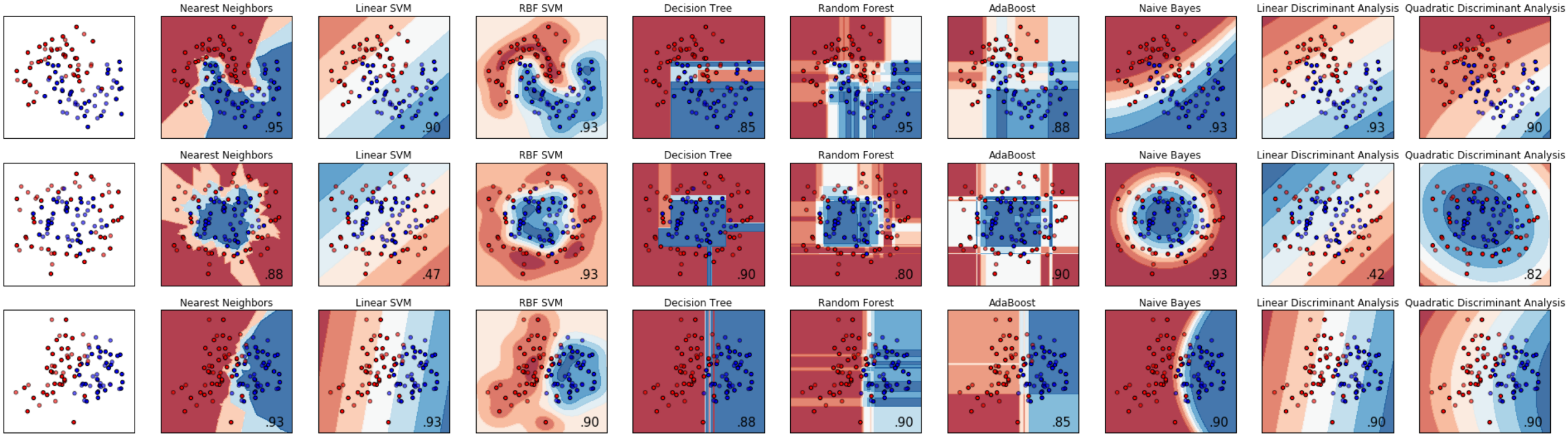
Red -> Testing Data
Blue -> Training Data

$$E = \frac{1}{k}\sum_{i=1}^{k} E_i(\mathbf{w})$$



Comparing Multiple Classifiers

Multiple Classifiers Across Multiple Data Sets



http://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf

Comparing Multiple Classifiers

- One versus One Across Multiple Data Sets
  - T-test: NO
  - Signed-Rank Wilcoxon
- Multiple Classifiers Across Multiple Data Sets
  - Friedman Test
  - Bonferroni-Dunn

Background for this Course

- As you will find out, many of the problems in pattern recognition will eventually be reduced to finding decision boundaries between classes
  - functions: curves, surfaces, planes, hyperplanes, hypersurfaces (in high dimensional spaces)
  - E.g., $\mathbf{y} = \mathbf{A}\mathbf{x}$
  - Need to solve matrix operations! (i.e., *linear algebra*)
- On the other hand, all of our work will be hindered by many sources of uncertainty
  - Limited data, noise in the measurement, or sampling from a process
  - Such uncertainty needs to be accounted (i.e., *probability*)

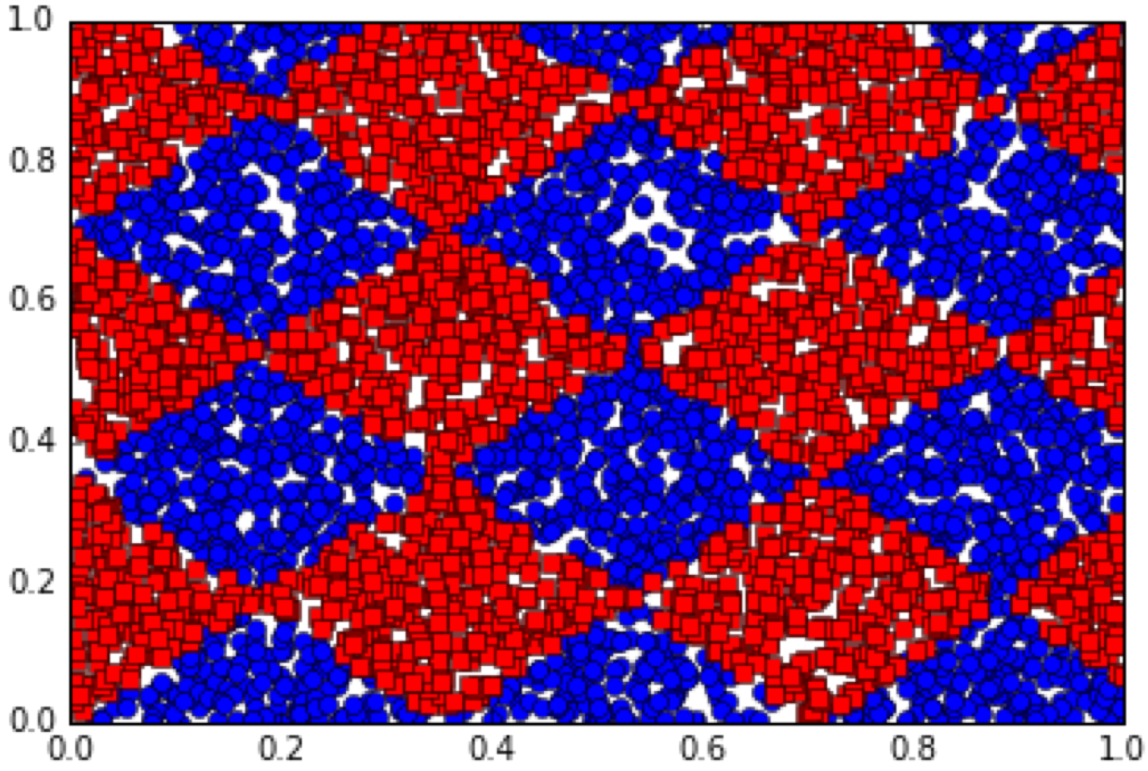
Data, Data, Data…

- Synthetic Data
  - Data that are not “real” but generate in a controlled environment to evaluating an algorithm
  - E.g., Multi-variate Gaussian data, checkerboard, Bernoulli, etc.
- Real-World Data
  - Data are collected from a survey or study. Noise, labeling errors, and closed form relationships are generally not known
  - Sources
    - UCI Machine Learning Repo (http://archive.ics.uci.edu/ml/)
    - Kaggle Data Sets (https://www.kaggle.com/datasets)

Generating a Checkerboard

In Code

```
def gen_cb(N, a, alpha):
    """
    N: number of points on the checkerboard
    a: width of the checker board (0<a<1)
    alpha: rotation of the checkerboard in radians
    """
    d = np.random.rand(N, 2).T
    d_transformed = np.array([d[0]*np.cos(alpha)-d[1]*np.sin(alpha),
                              d[0]*np.sin(alpha)+d[1]*np.cos(alpha)]).T
    s = np.ceil(d_transformed[:,0]/a)+np.floor(d_transformed[:,1]/a)
    lab = 2 - (s%2)
    data = d.T
    return data, lab
```

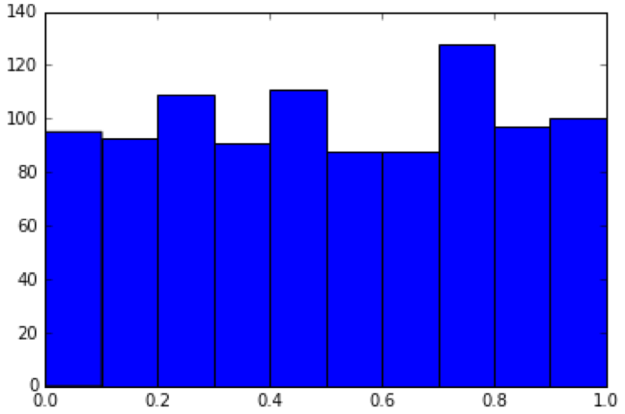



```
X, y = gen_cb(5000, .25, 3.14159/4)
plt.figure()
plt.plot(X[np.where(y==1)[0], 0], X[np.where(y==1)[0], 1], 'o')
plt.plot(X[np.where(y==2)[0], 0], X[np.where(y==2)[0], 1], 's', c = 'r')
```

Generating Data from Probability Distributions

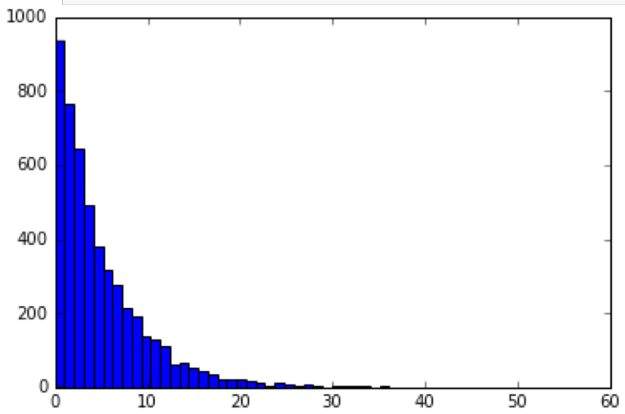
Uniform

```
x = np.random.rand(1000)
d = plt.hist(x)
print "The mean of x is ", np.mean(x)
```



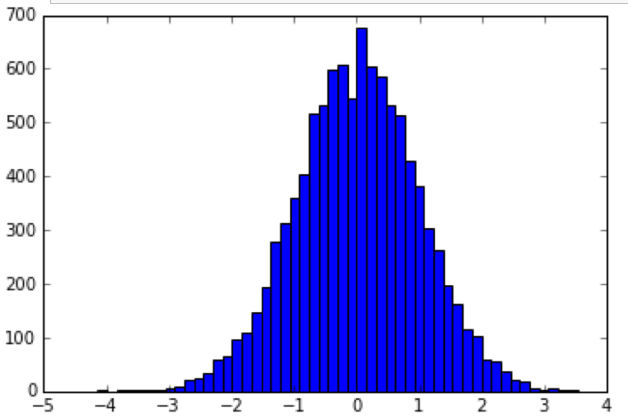
Exponential $f(x;\beta) = \frac{1}{\beta}\exp\left(-\frac{x}{\beta}\right)$

```
beta = .2
x = np.random.exponential(1/beta, 5000)
d = plt.hist(x, bins = 50)
print "The mean of x is ", np.mean(x)
```



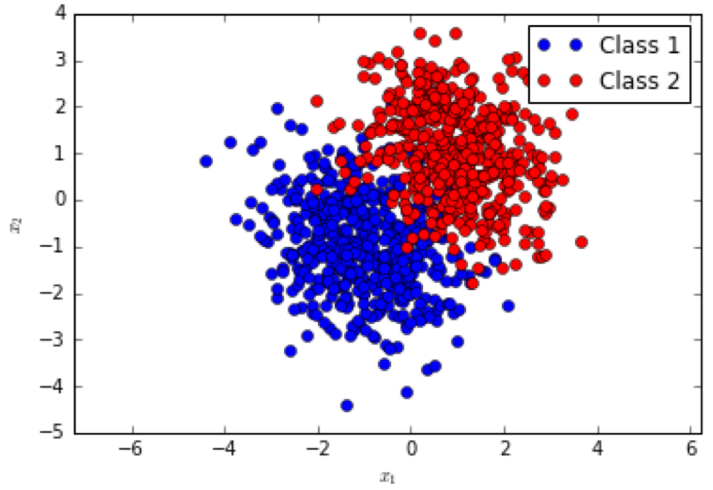
Gaussian $f(x;\mu;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\,\sigma^2}\right)$

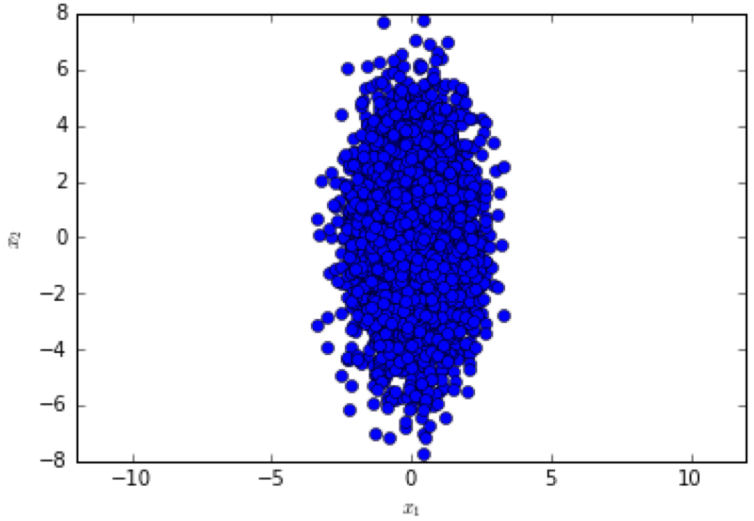
```
x = np.random.randn(10000)
d = plt.hist(x, bins=50)
print "The mean of x is ", np.mean(x)
```

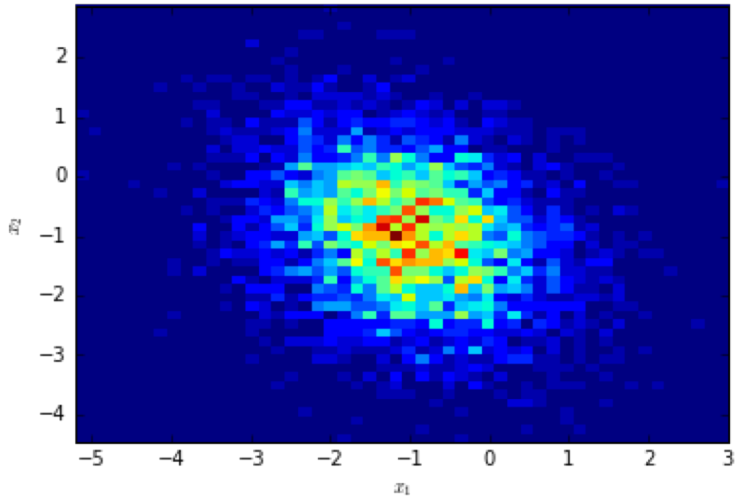


Generating Data from Probability Distributions

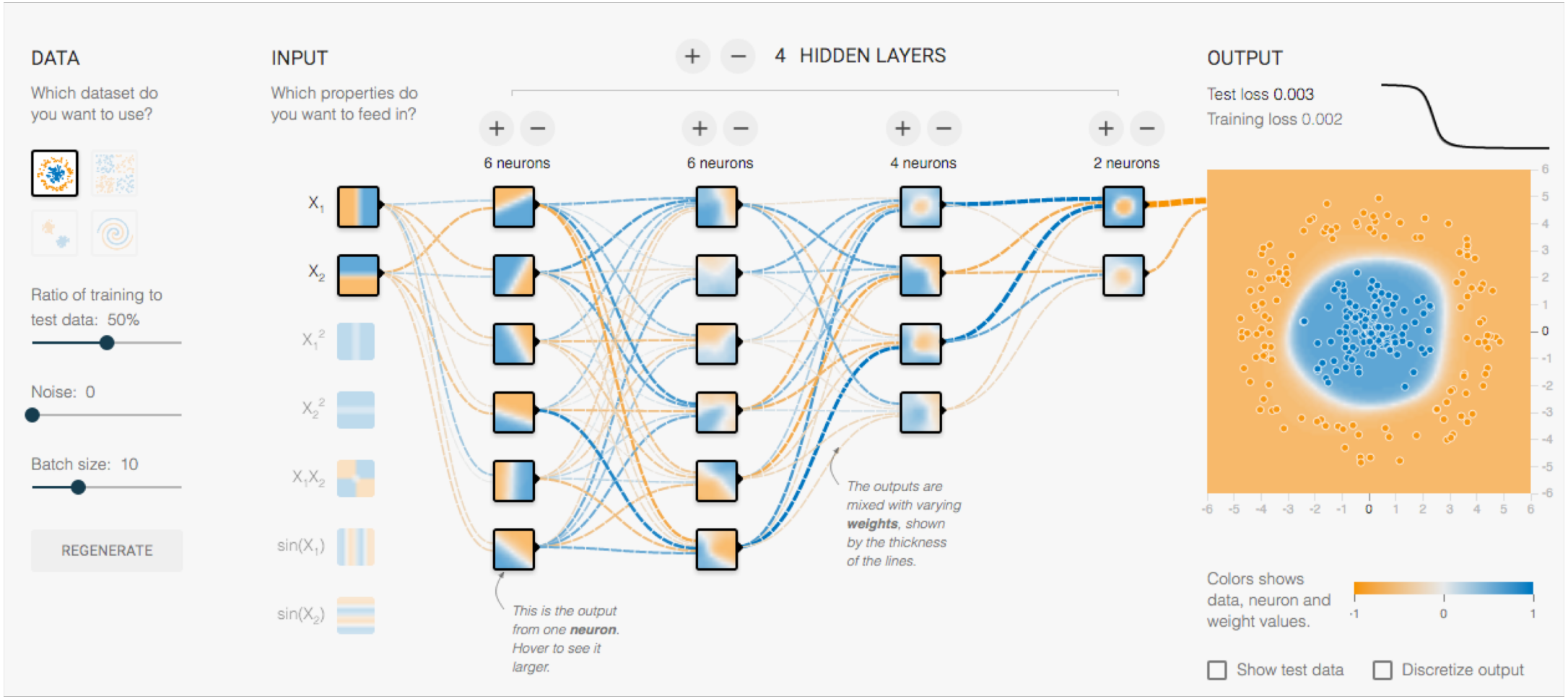
```
x = np.random.multivariate_normal([-1, -1], [[1, -.25], [-.25, 1]], 500).T
y = np.random.multivariate_normal([1, 1], [[1, -.25], [-.25, 1]], 500).T
plt.plot(x[0], x[1], 'o', c='b')
plt.plot(y[0], y[1], 'o', c='r')
plt.axis('equal')
plt.xlabel('$x_1$')  # use latex in the figure axis labels
plt.ylabel('$x_2$')
plt.legend(("Class 1", "Class 2"))
plt.show()
```







Meet Google's ML Library



http://playground.tensorflow.org/

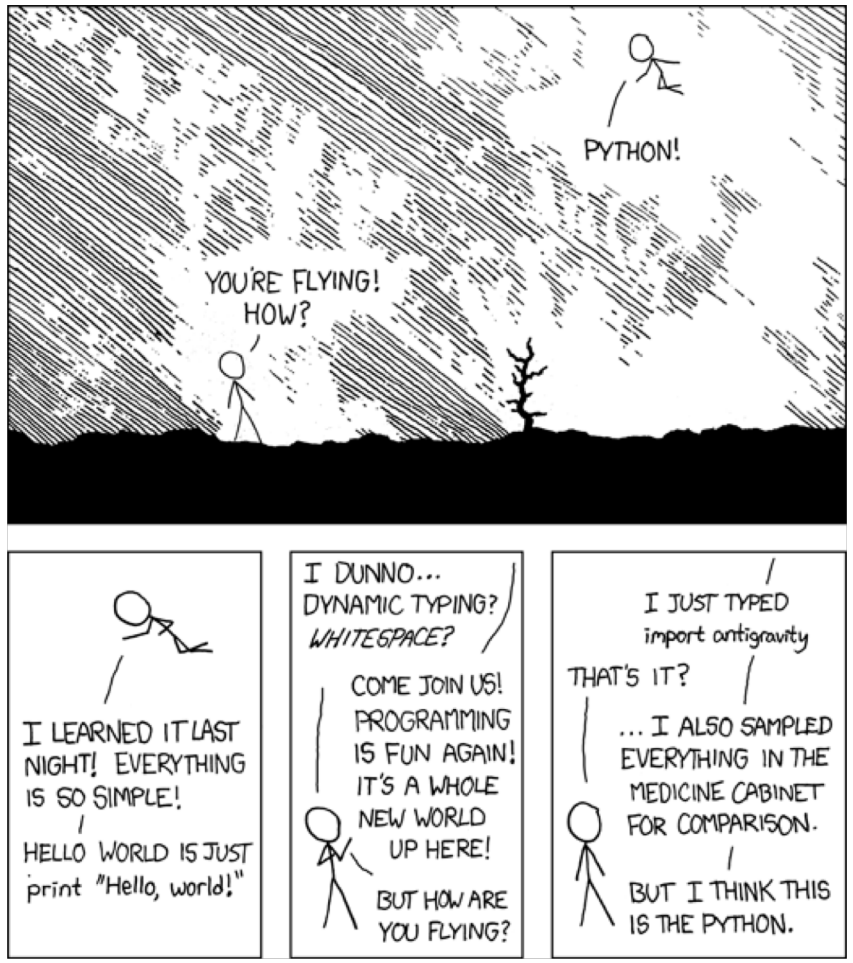
Mini-Assignment

- Acclimate yourself with Python, IPython, and Scikit-Learn
  - Code will be posted; however, you should try to do this on your own
- Generate data using the built in random number generators in Numpy
- Read Chapters 1 & 2 (19 wouldn’t hurt either)

Crash Course in Python

What is Python?

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics
  - Similar to Matlab
  - Many 3rd party tools
  - Open source and heavily supported
- Anecdotal evidence suggests that one Python programmer can finish in two months what two C++ programmers can't complete in a year



https://developers.google.com/edu/python/

Why Python

- Python is consistently ranked as one of the top programming languages to know and the salaries support this claim
- Developing code in Python is fast and easy to develop compared to other languages such as C++ and Java
- The developers have a sense of humor

```
>>> import antigravity
```

- It is free! Numpy and Scipy implement much of Matlab's base functionality

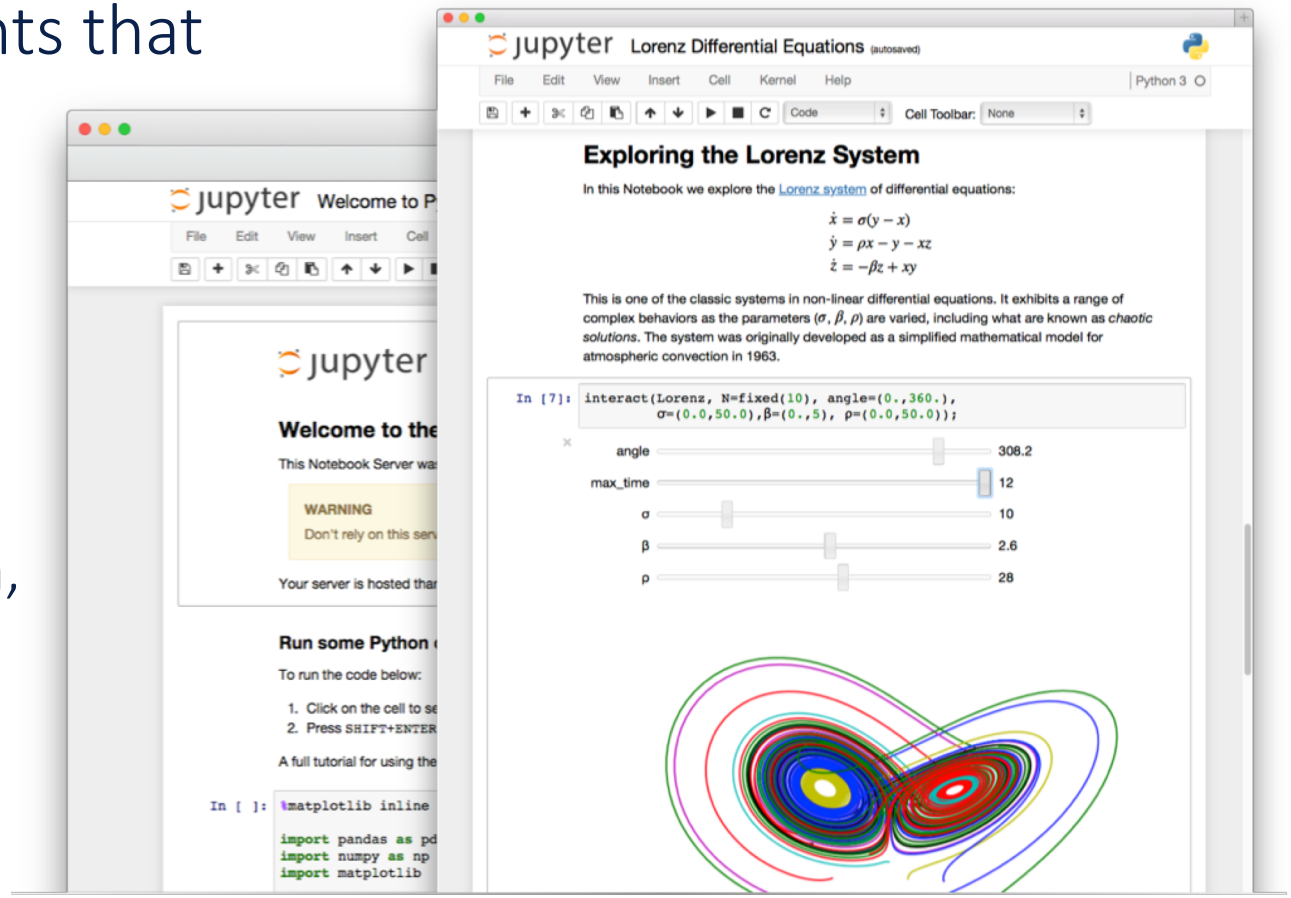
Google Python Class

with Nick Parla

Day 1 Part 1
Introduction and Strings

Interactive Python (now Jupyter)

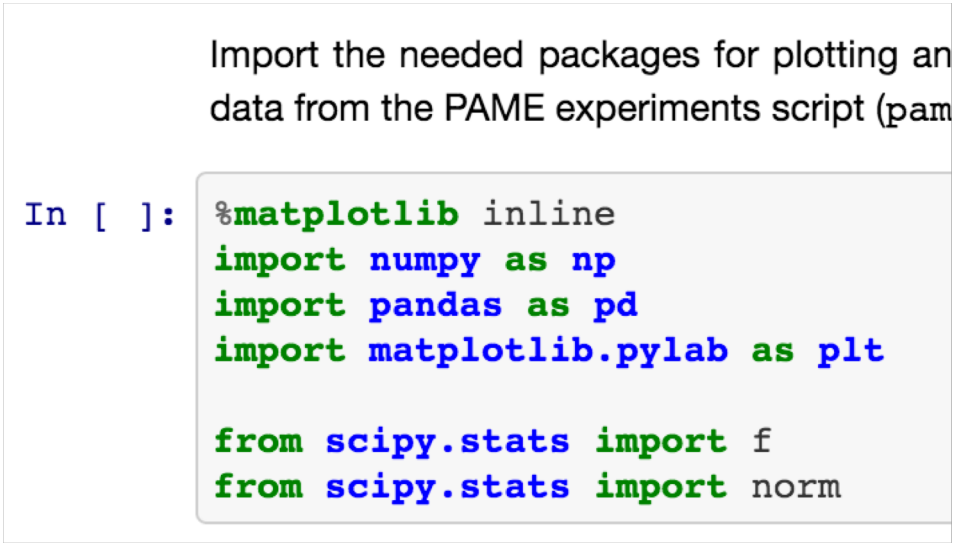
- The Jupyter Notebook is a web application that allows you to create documents that contain live code, equations, and visualizations
  - data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.
  - supports multiple languages: Python, Lua, Julia, R, …



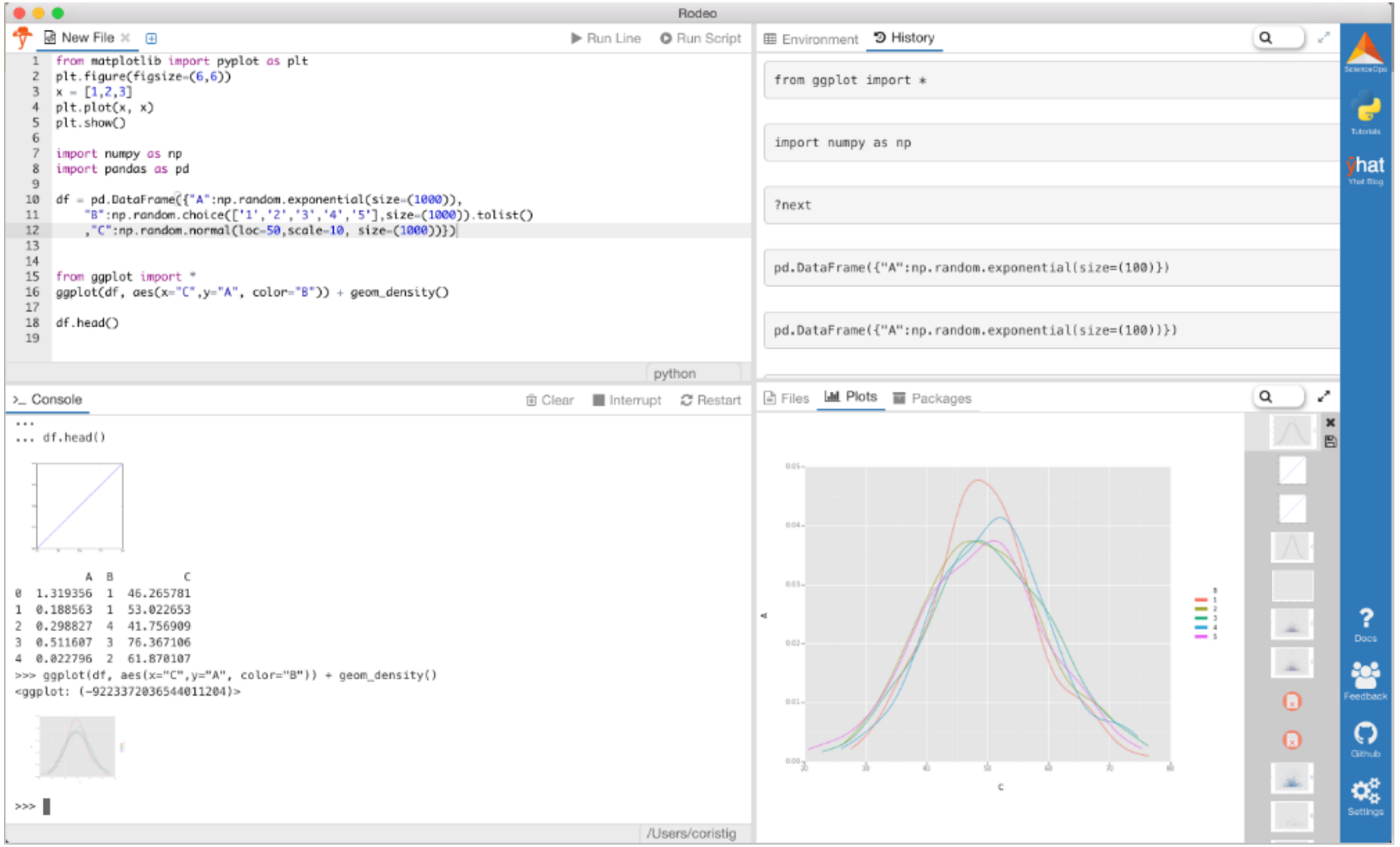
Invoking Jupyter

- At the shell, move to the directory where you want to run your project then
  `$ ipython notebook`

- Set matlplotlib to be inline graphics
  - Cells: Code, Shell or Markdown



Rodeo (Gentle Transition Away from Matlab)



https://www.yhat.com/products/rodeo

Pandas

- Library for implementing data structures that are designed to ease of data manipulation
  - Essentially implements R's data frame
  - Munging, cleaning, modeling, analyzing and organizing data

- Nice for exploratory data analysis

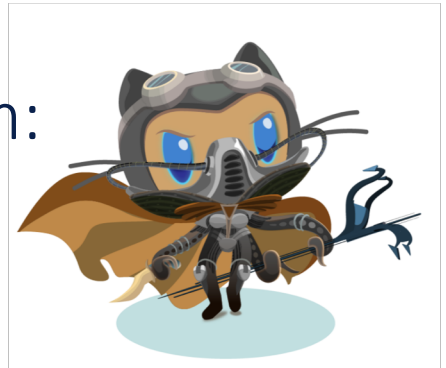
http://pandas.pydata.org/

Tensorflow

- TensorFlow is an open source software library for numerical computation using data flow graphs
  - Free & developed by Google
  - Lots of examples
  - Used in practice
- The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop
- We will use TensorFlow later in the course for artificial neural networks and deep learning

IDEs & Editors

- VS Studio code (recommended) https://code.visualstudio.com/
- Sublime Text 2: http://www.sublimetext.com/
- Atom: https://atom.io/
- PyCharm: https://www.jetbrains.com/pycharm/
- Vim:



Miscellaneous

- yhat's ggplot: Alternative to Matplotlib (R's ggplot2)
- Matplotlib: Plotting library with similar commands as Matlab
- Stan: Bayesian inference package
- Scikit-XYZ: learn, optimization, bioinformatics, image processing, etc.
- Anaconda: USE THIS!!!! It installs a version of Python that has a bunch of packages. Works out of the box.

Suggested Exercises

- Google's Python Crash Course: https://developers.google.com/edu/python/

Sharing Code and Data for the Course

- I will be using Github to host the code that I develop and demo for this course. Feel free to use it and make suggestions to change the content as the course progresses

https://github.com/gditzler/UA-ECE-523-Sp2018