

# Machine Learning



what society thinks I  
do



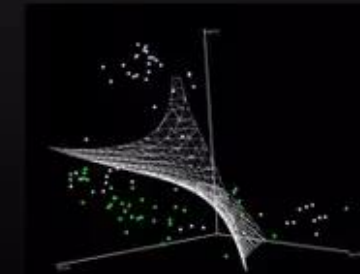
what my friends think  
I do



what my parents think  
I do

$$\begin{aligned} L_p &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_i \alpha_i \\ \alpha_i &\geq 0, \forall i \\ \mathbf{w} &= \sum_i \alpha_i y_i \mathbf{x}_i, \sum_i \alpha_i y_i = 0 \\ \nabla_{\hat{\theta}} \ell(\theta_t) &= \frac{1}{n} \sum_{i=1}^n \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t), \\ \theta_{t+1} &= \theta_t - \eta_t \nabla \ell(x_{i(t)}, y_{i(t)}; \theta_t) - \eta_t \cdot \nabla r(\theta_t) \\ \mathbb{E}_{i(t)}[\ell(x_{i(t)}, y_{i(t)}; \theta_t)] &= \frac{1}{n} \sum_i \ell(x_i, y_i; \theta_t). \end{aligned}$$

what other programmers  
think I do



what I think I do

```
>>> from scipy import svm
```

what I really do

INFO 251: Applied Machine Learning

## Intro to Machine Learning

# Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- Machine Learning
  - **Design of Machine Learning Experiments**
  - Linear Models and Gradient Descent
  - Non-linear models
  - Neural models
  - Unsupervised Learning
  - Practicalities, Fairness, Bias
- Special topics

# Today's Outline

- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning experiments

# Key Concepts (today's lecture)

- Representation
- Evaluation
- Optimization
- Supervised Learning
- Unsupervised Learning
- The curse of dimensionality
- Feature engineering
- Overfitting
- Generalization
- Cross-validation
- Bootstrap
- Accuracy, ROC, AUC, F-scores
- Baselines
- Error analysis
- Ablative analysis

# Machine Learning: Introduction

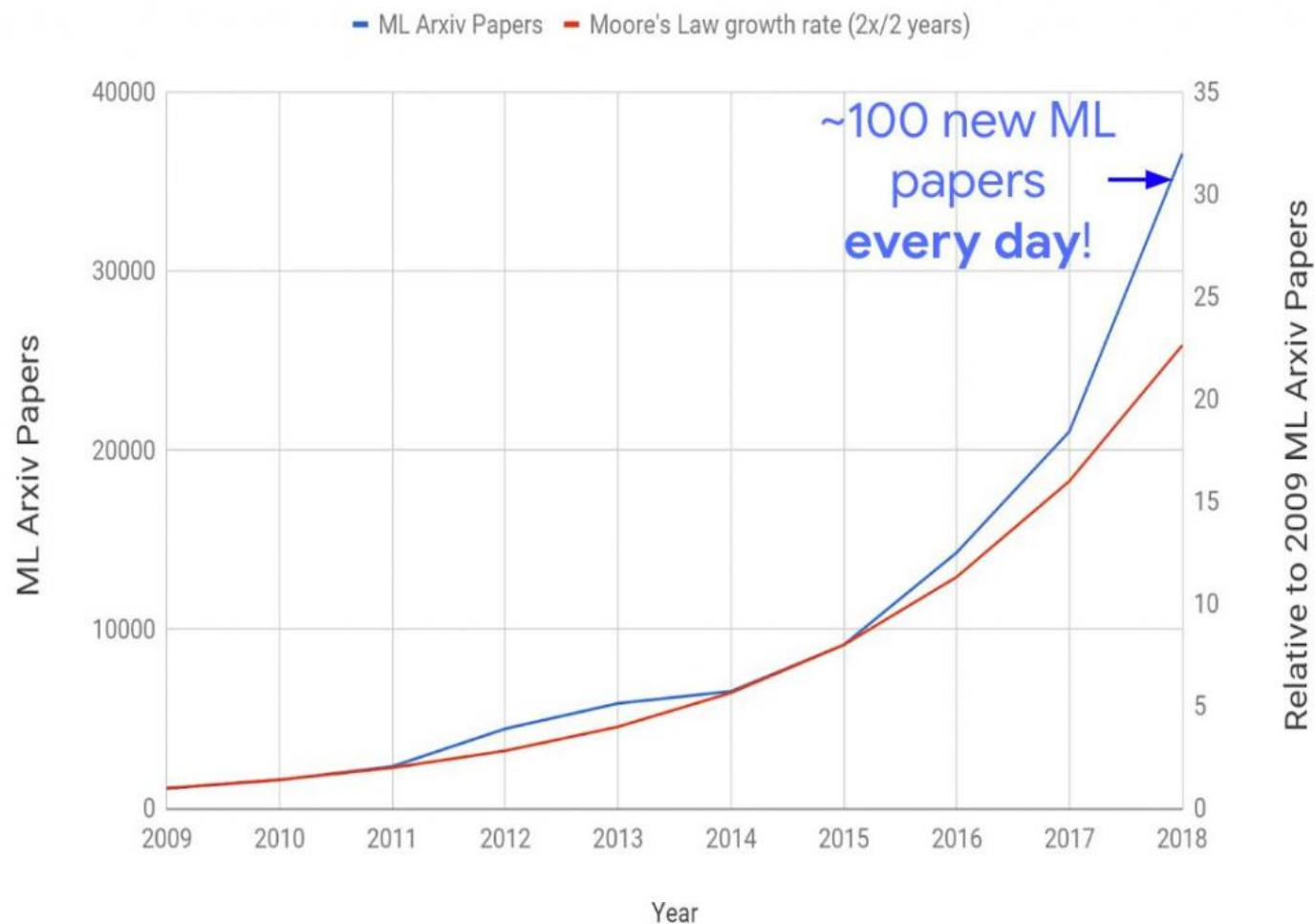
- What is machine learning?



- **Machine learning** is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Such algorithms operate by building a model based on **inputs** and using that to make **predictions or decisions**, *rather than following only explicitly programmed instructions*.

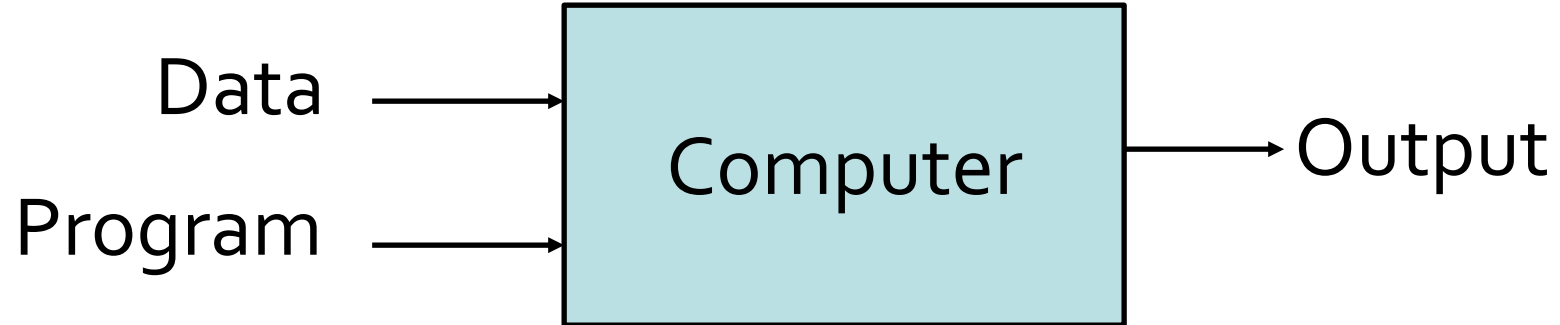
# Machine Learning: Context

## Machine Learning Arxiv Papers per Year

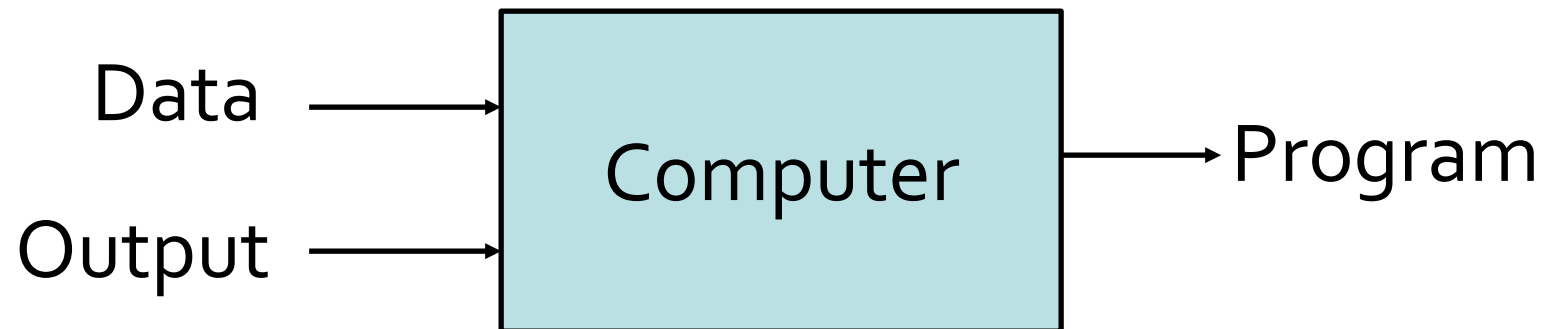


# What is Machine Learning?

- Traditional Programming



- Machine Learning



# What is Machine Learning?

- Econometrics: We start with a model  $f(\cdot)$  of how the world works, e.g.,  $Y = \alpha + \beta X + \epsilon$ 
  - Our focus is on unbiased (and therefore generalizable) estimation of  $\hat{\beta}$
  - How we specify  $f(\cdot)$  is critical – the validity of causal inferences about  $\beta$  depend on it
- Machine learning: We start with a model  $f(\cdot)$ 
  - Our focus is on accurate (and generalizable) predictions of  $\hat{Y}$
  - This opens the door to new families of models that optimize for  $\hat{Y}$ , often at the expense of interpretability



# Two paradigms of regression

1. Statistics / Econometrics
  - Explaining relationships
  - Understanding causality
  - E.g.: Why do customers churn?
2. Machine Learning / Computer Science
  - Predicting the future
  - Extracting generalizable patterns
  - E.g., Which customers will churn?

# ML in a Nutshell

- Tens of thousands of ML algorithms exist
- Every ML algorithm has three components:
  1. **Representation** (i.e., the Model)
  2. **Evaluation** (i.e., an objective function)
  3. **Optimization** (e.g. Search)

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

# Representation / Model

- “Choosing a representation for a learner is tantamount to choosing the set of classifiers that it can possibly learn. This set is called the *hypothesis space* of the learner”
  - Decision trees
  - Instance-based
  - Neural networks
  - Support vector machines
  - [Probabilistic (graphical) models]
  - Model ensembles
  - ...
- It's common to fixate on the model representation, but in practice, many other factors are more important

# Evaluation

- Is our model effective?

- “Coefficient of determination”:  $R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$ 
  - i.e., residual sum of squares / total sum of squares
  - “fraction of explained variance”
- Accuracy
- Precision, Recall, F-scores, Area under the Curve
- Squared error, RMSE, MAE
- (Log-) Likelihood
- Cost / Utility
- Entropy, K-L divergence, etc.

# Optimization / Search

- How to improve?
  - Combinatorial optimization (discrete)
    - E.g.: Greedy search
  - Convex optimization (continuous)
    - E.g.: Gradient descent
  - Constrained optimization
    - E.g.: Linear programming

# Outline

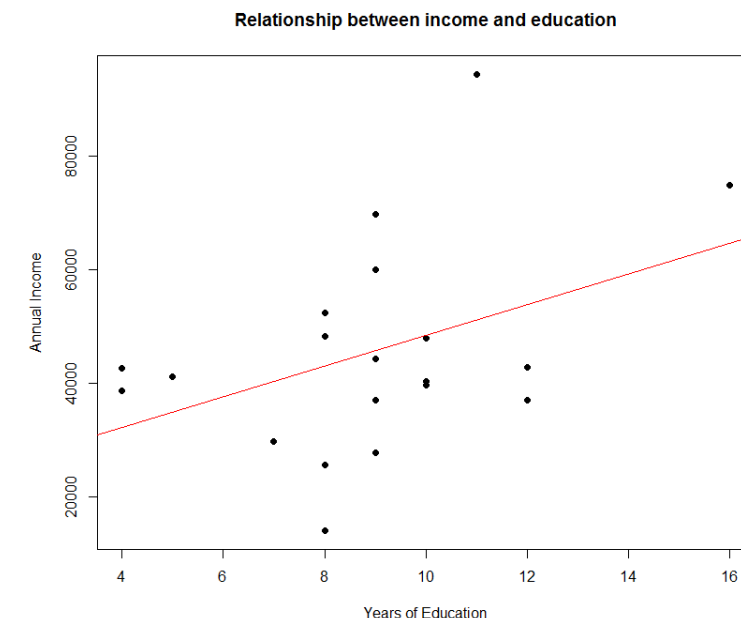
- Introduction to Machine Learning
- **Supervised vs. Unsupervised Learning**
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning Experiments

# Supervised vs. Unsupervised

- Key distinction:
  - Whether or not you know the “right” answer

# Supervised Learning

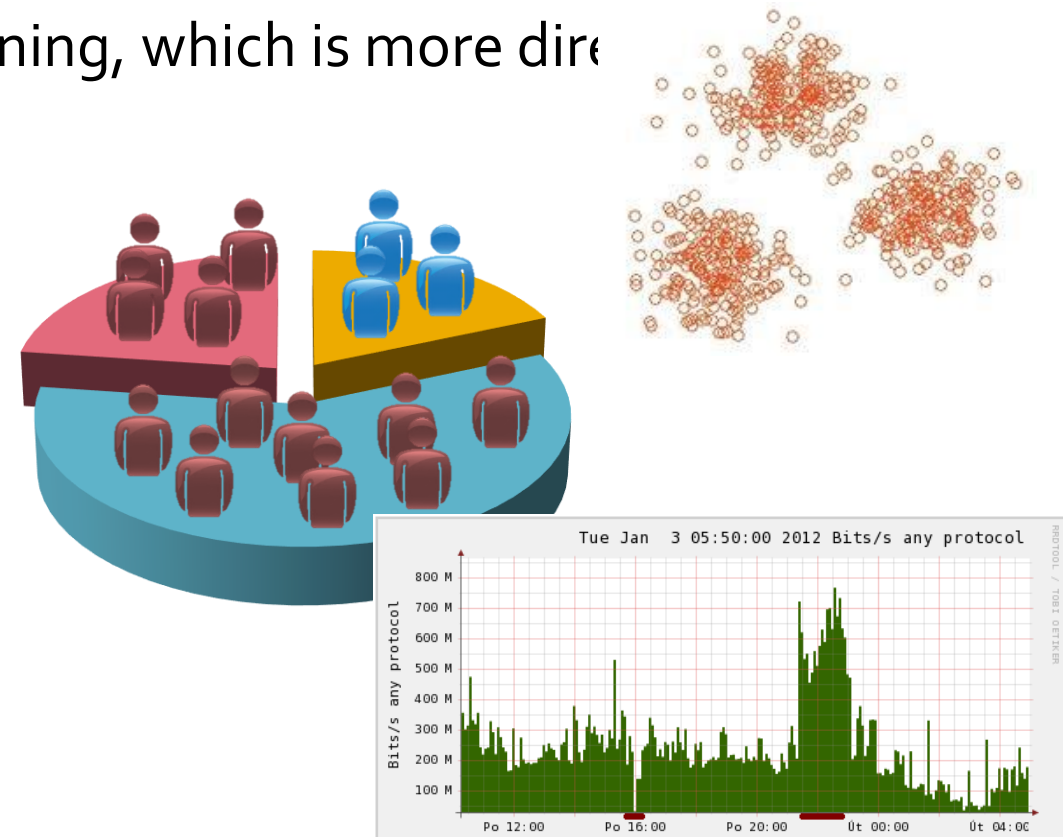
- We know the “right answer” for some values
  - Goal is typically to model relationship between inputs output, where values of output are known
- Examples
  - Disease classification, credit scoring, etc.
- Methods:
  - Linear models (regression, logistic regression, SVM)
  - Decision Trees, random forests
  - Neural Networks
  - Ensemble methods





# Unsupervised Learning

- We don't know the "right answer", the right groupings, "ground truth"
  - Goal is typically to discover underlying structure in the data
  - Often more exploratory than supervised learning, which is more direct
- Examples
  - Market segmentation, disease classification
  - Visualizing complex data
- Methods:
  - K-means and hierarchical clustering
  - Principal Component analysis
  - SVD, NMA, LDA



# Other approaches to ML

- Semi-supervised learning
  - We have some labeled instances
- Reinforcement learning
  - Learning by interacting with an environment
  - Rewards from sequence of actions
- Etc.
  - Fair\* Machine Learning
  - Online learning
  - Adversarial learning
  - ...

# Outline

- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- **Key Issues in Machine Learning**
- Design of Machine Learning Experiments

# Key Issues in (Supervised) Machine Learning

- Generalization
  - “The fundamental goal of machine learning is to generalize beyond the examples in the training set. This is because, no matter how much data we have, it is very unlikely that we will see those exact examples again at test time.”
- ... and Overfitting
  - (More on this soon)



Thanks to machine-learning algorithms,  
the robot apocalypse was short-lived.

# Key Issues in (Supervised) Machine Learning

- Fast vs. exact solutions



# Key Issues in (Supervised) Machine Learning

- Feature engineering
  - “Easily the most important factor is the features used.”
  - “This is typically where most of the effort in a machine learning project goes. It is often also one of the most interesting parts, where intuition, creativity and “black art” are as important as the technical stuff.”

# Key Issues in (Supervised) Machine Learning

- More Data Matters
  - “As a rule of thumb, a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it..”

# Key Issues in (Supervised) Machine Learning

- Ensembles work
  - Bagging: resample the training data to generate multiple data sets, and train classifiers on each one
  - Boosting: Focus on examples that are hard to learn
  - Stacking: Use models to learn from the outputs of other models



# Key Issues in (Supervised) Machine Learning

- Interpretability is (usually) important
  - There is beauty in simplicity!
  - Interpretability is hard to measure, but often trumps other measures of performance

# Key Issues in (Supervised) Machine Learning

- Summary
  - Generalization and overfitting
  - Feature engineering
  - More data matters
  - Ensembles work
  - Interpretability is important

# Outline

- Introduction to Machine Learning
- Supervised vs. Unsupervised Learning
- Key Issues in (Supervised) Machine Learning
- Design of Machine Learning experiments
  - **Motivation**
  - Training, testing, validation, cross-validation and bootstrap
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

# What model should you use?

- “All models are wrong but some are useful.”



George Box  
1919 - 2013

# What model should you use?

- Which of the following models is the “right” one?

```
sm.ols('income ~ education', data=s1).model.fit().summary()
```

```
(Intercept)    education
    24287.71         2518.60
```

```
sm.ols('income ~ education + youngkids', data=s1).model.fit().summary()
```

```
(Intercept)    education    youngkids
    24590.811         2565.921    -2383.692
```

```
sm.ols('income ~ education + youngkids + age', data=s1).model.fit().summary()
```

```
(Intercept)    education    youngkids         age
12013.36940    2660.38919         19.25572    274.02269
```

- “But doesn’t R-squared tell us the best model?”

```
sm.ols('income ~ education', data=s1).model.fit().summary().rsquared
```

```
0.1066281
```

```
sm.ols('income ~ education + youngkids', data=s1).model.fit().summary().rsquared
```

```
0.1104819
```

```
sm.ols('income ~ education + youngkids + age', data=s1).model.fit().summary().rsquared
```

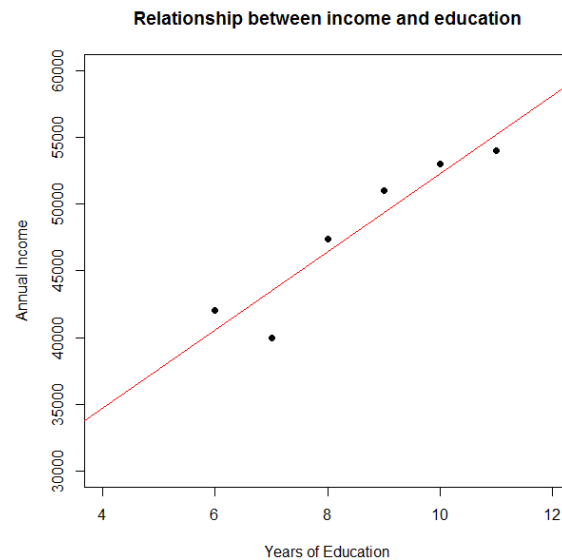
```
0.1214696
```

```
sm.ols('income ~ education + youngkids + age + random_noise', data=s1).summary().rsquared
```

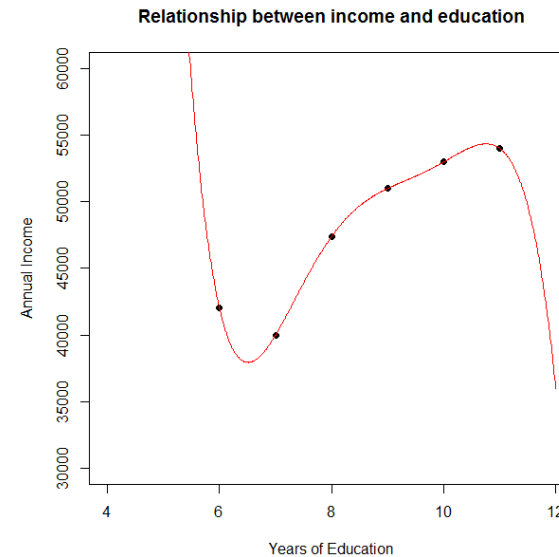
```
0.1291423
```

# Generalization and Overfitting

- Overfitting: When a model fits the training set very well (e.g., high  $R^2$ ) but fails to generalize to new data



$$wages_i = \alpha + \beta * educ_i + error_i$$



$$wages_i = \alpha + \beta_1 * educ_i + \dots + \beta_5 * educ_i^5 + error_i$$

# Generalization and Overfitting

- $R^2$  does not tell you which model is “right”
- Our  $R^2$  increases as we
  - add complexity
  - iterate on features
  - try different models
  - use different datasets
- **Good fit is not the same as a good model!**

# So... What model should you use?

- “All models are limited by the validity of the assumptions on which they ride.”
- “Assumptions behind models are rarely articulated, let alone defended.”

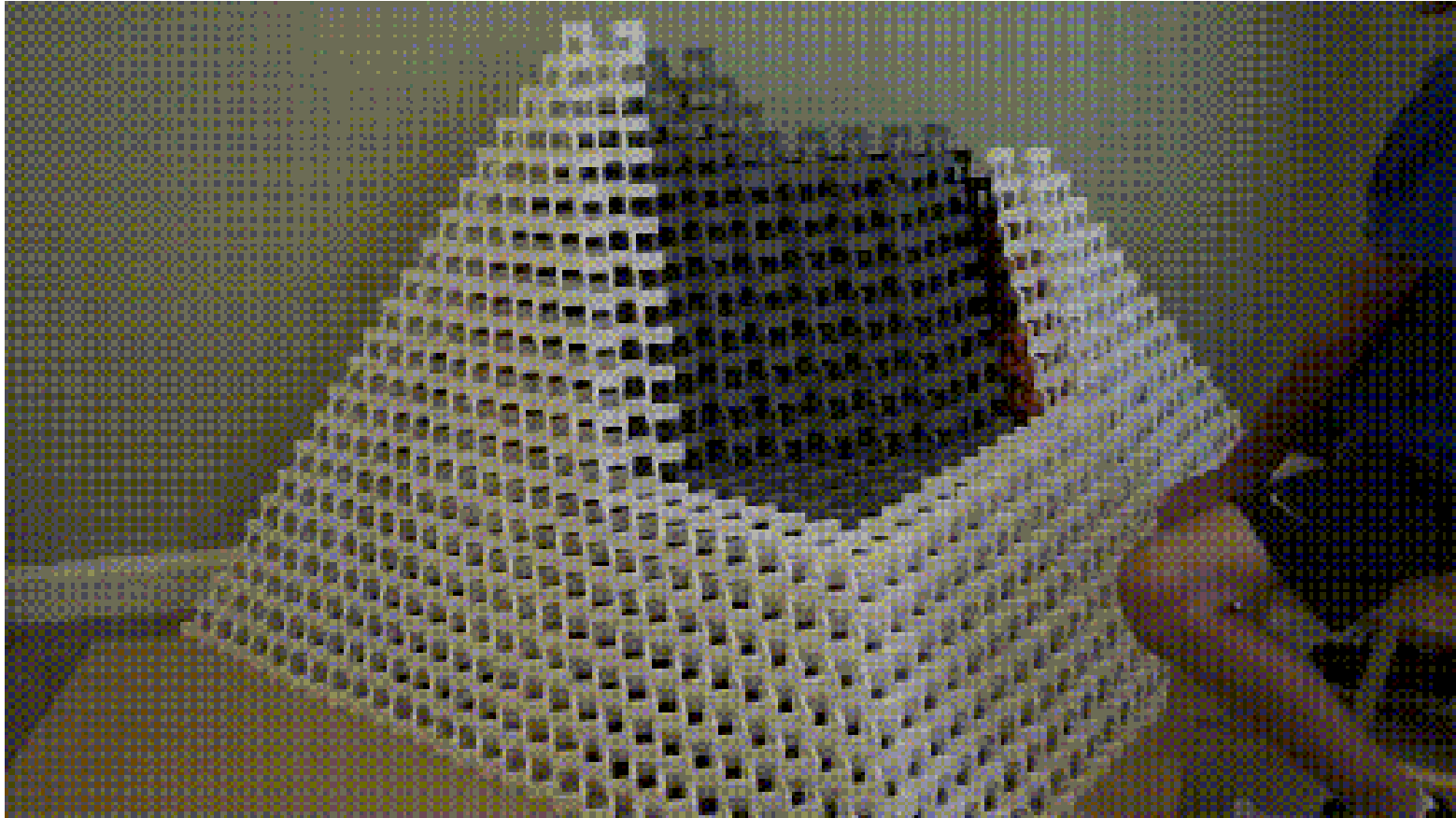


David Freedman  
1938 - 2008



# What model should you use?

- This motivates a more principled way to select a model...

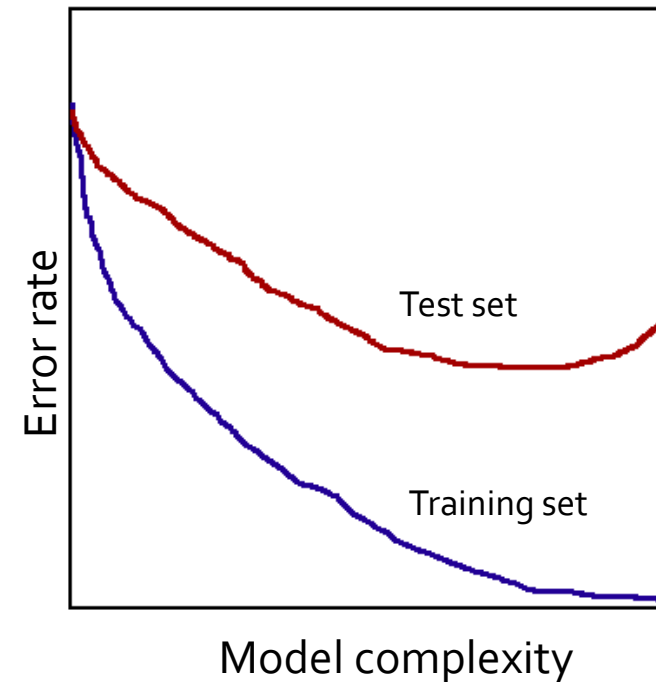


# Outline

- Design of Machine Learning experiments
  - Motivation
  - **Training, testing, validation, cross-validation and bootstrap**
  - Measuring performance
  - Choosing appropriate baselines
  - Error Analysis

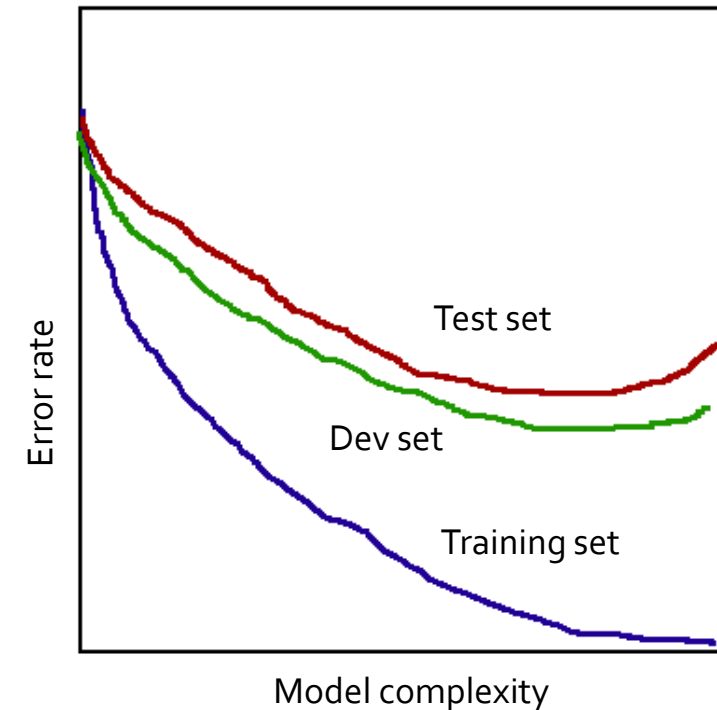
# Training and Testing

- ML experiments typically separate data into a **training set** and a **testing set**
  - Model is fit on training set
  - Performance is measured on test set



# Validation (development) data

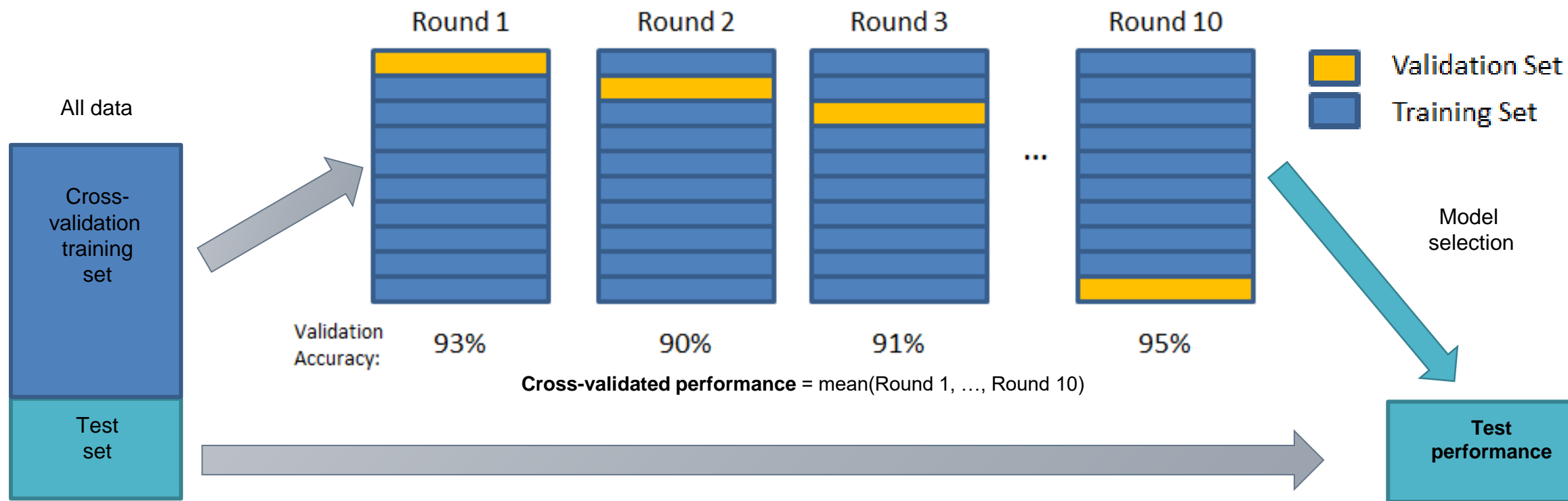
- Splitting into training + testing is often not enough
  - Each time you look at the test set, you introduce bias (in yourself!)
  - Hyperparameters must be chosen
  - Model selection, feature selection, etc.
- Validation/development data
  - A third split of the data
  - Used as a pseudo-test set for hyperparameter tuning
- Measure and report performance on test set
  - Unseen until very end



# Cross-Validation

- Given unlimited data, it's easy to find new test data. But what if you have limited data?
  - Your “random” sample of training data may not be representative
- $k$ -fold cross-validation
  - Randomly partition data into  $k$  equal size subsamples
  - Use each of  $k$  folds as validation set once
  - Average performance across  $k$  test runs – this is your “CV test performance”
- Test performance
  - Final model performance evaluated on a separate test set that was *never used in cross-validation*

# Cross-Validation



# Cross-Validation

- Stratified cross-validation
  - Select folds so that the mean response value is approximately equal in all the folds, or so that some other parameter is balanced across folds
- Leave-one-out
  - Special case where  $K = N$
  - Pro: deterministic, almost all data used each fold
  - Con: computationally intensive
  - Con: Can't stratify, can overfit
    - Calculate error rate for single-feature binary classifier with evenly split positive/negative instances

# Bootstrap





- Cross-Validation
  - Partitioning of data into  $k$  folds means each instance is used exactly one (either as train or test)
- Bootstrap
  - Instead, sample *with replacement* from data
  - Unsampled data become the validation set
  - Training data: 63.2% unique; validation: 36.8%
    - Probability that an instance is not picked =  $1 - (1/n)$
    - $\left(1 - \frac{1}{n}\right)^n \cong e^{-1} = 0.368$



# Outline

- Design of Machine Learning experiments
  - Motivation
  - Training, testing, validation, cross-validation and bootstrap
  - **Measuring performance**
  - Choosing appropriate baselines
  - Error Analysis

# Evaluating Classifiers

		YOU CONCLUDE	
		Effective	No Effect
THE TRUTH	Effective		Type II Error (low power) 
	No Effect	Type I Error (5% of the time) 	

- Accuracy, Precision, Recall, and F-scores
  - Accuracy:  $(tp + tn) / (tp + tn + fp + fn)$
  - Recall:  $tp / (tp + fn)$  [% of real positives identified]
  - Precision:  $tp / (tp + fp)$  [% of positives that are correct]
  - F stat:  $2 * (P * R) / (P + R)$

# Evaluating Classifiers: Other metrics

- There's more to the story than accuracy...

	Accuracy	Recall	Precision	F	AUC	% Answered Yes
<i>Panel A: Assets and Housing</i>						
Owens a radio	0.976	1.000	0.976	0.988	0.899	0.973
Owens a bicycle	0.676	0.552	0.678	0.609	0.722	0.456
Household has electricity	0.819	0.533	0.761	0.627	0.828	0.285
Owens a television	0.855	0.497	0.738	0.594	0.814	0.214
Has indoor plumbing	0.887	0.250	0.842	0.386	0.843	0.142
Owens a motorcycle/scooter	0.899	0.011	1.000	0.022	0.772	0.102
Owens a car/truck	0.945	0.213	0.867	0.342	0.849	0.068
Owens a refrigerator	0.954	0.180	1.000	0.305	0.878	0.055
Has landline telephone	0.992	0.125	1.000	0.222	0.562	0.009
<i>Panel B: Social Welfare Indicators</i>						
Hospital bills in last 12 months	0.633	0.890	0.633	0.740	0.653	0.587
Very ill in last 12 months	0.686	0.188	0.550	0.280	0.671	0.325
Death in family in last 12 months	0.665	0.183	0.632	0.284	0.619	0.363
Flood or drought in last 12 months	0.788	0.086	0.607	0.151	0.706	0.219
Fired in last 12 months	0.901	0.022	1.000	0.043	0.731	0.101

Table 2: Model performance at predicting responses from survey respondents based on call records data

# Evaluating Classifiers: ROC Curves

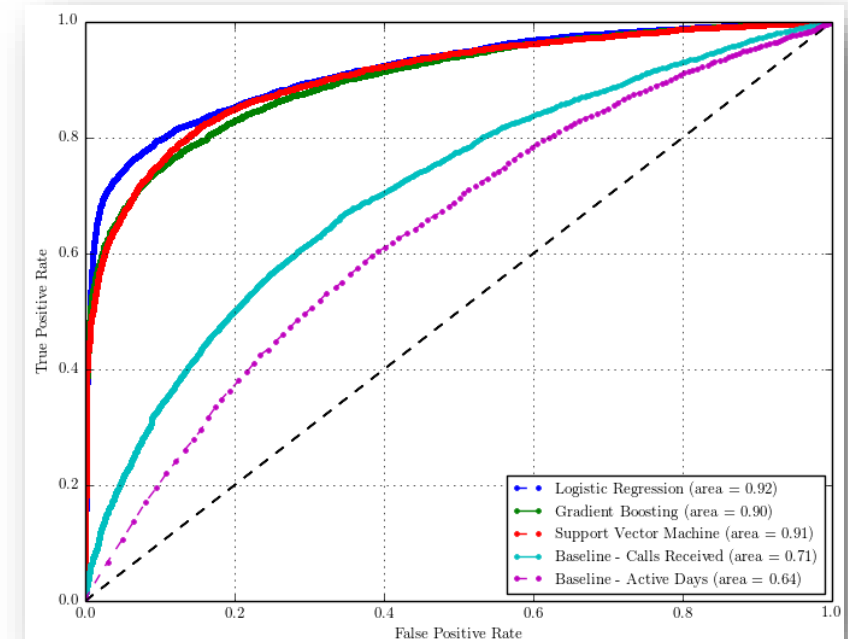
- Classifiers use thresholds on predicted values
  - e.g. the output of a logistic regression
- What is the optimal threshold?
  - High threshold
    - Few false positives 😊
    - Many false negatives ☹️
  - Low threshold
    - Many false positives ☹️
    - Few false negatives 😊

# Evaluating Classifiers: ROC Curves

- “Area Under Curve”:
  - equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one:

$$A_{ROC} = \int_0^1 \frac{TP}{P} d\frac{FP}{N} = \frac{1}{P N} \int_0^N TP dFP$$

- Independent of specific threshold



# Evaluating Numeric Predictions

- (Root) Mean-squared error:
  - Mean absolute error:
  - Correlation coefficient
  - etc. -- these tend to be highly correlated

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2.$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|.$$

- “Goodness of fit”
  - Coefficient of determination:  $R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$ 
    - Residual sum of squares / total sum of squares
    - “fraction of explained variance”
  - F-statistics (overall significance of model)
  - Chi-squared (for categorical data)

# Outline

- Design of Machine Learning experiments
  - Motivation
  - Training, testing, validation, cross-validation and bootstrap
  - Measuring performance
  - **Choosing appropriate baselines**
  - Error Analysis

# Experimental process: baselines

- We want to quantify progress relative to something meaningful:
  - vs. random guessing
  - vs. most likely label
  - vs. something simple and intuitive
  - vs. the current state of the art



# Baselines for POS tagging

Fruit flies like a banana

|       |       |       |       |

**adj   noun   verb   det   noun**

- Goal: label words with part of speech
- What are reasonable baselines?
  - Always predict "noun" : 13%
  - Predict most common tag for each word: 81%
  - Current state of the art: 97%

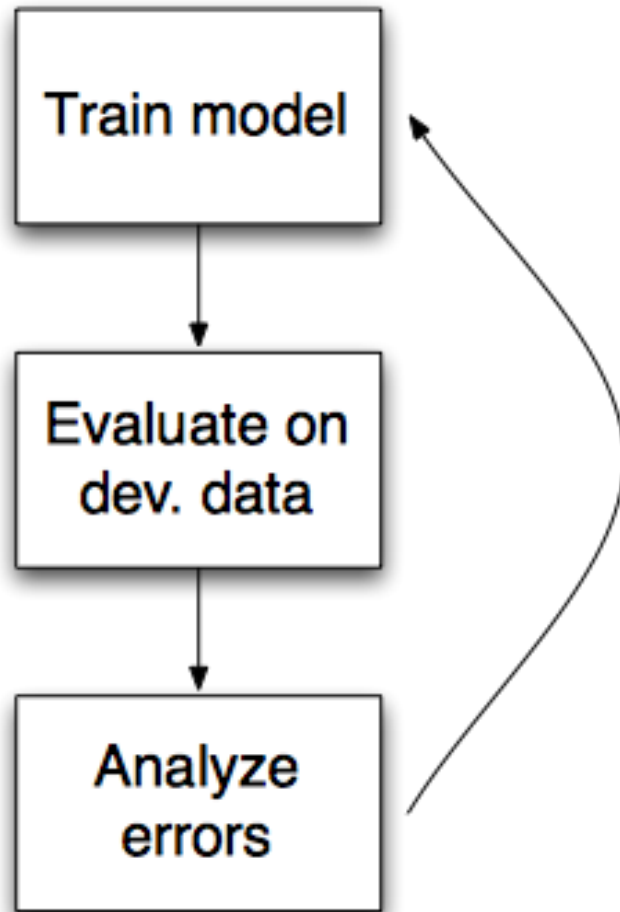
# Outline

- Design of Machine Learning experiments
  - Motivation
  - Training, testing, validation, cross-validation and bootstrap
  - Measuring performance
  - Choosing appropriate baselines
  - **Error Analysis**

# Error analysis

- **Error analysis** tries to explain the difference between current performance and perfect performance
  - One of the most under-appreciated steps, can make an immense difference in performance
  - Things rarely work out of the box
- **Ablative analysis** tries to explain the difference between some baseline (much poorer) performance and current performance

# Error analysis



- **Issue:** No tags for rare words ("prestidigitation")
- **Solution:** Add suffix features
- **Issue:** Spam classifier misclassifying lots of emails
- **Solution:** Add features that capture deliberate misspellings (c1alis, v1agra)

# Ablative Analysis

- E.g., Suppose that you've built a good anti-spam classifier by adding lots of clever features to logistic regression:
  - Spelling correction
  - Sender host features
  - Email header features
  - Email text parser features
  - Javascript parser
  - Features from embedded images
- Question: How much did each of these components really help?

# Ablative Analysis

- Simple logistic regression without any clever features: 94% performance
- Just what accounts for the improvement from 94 to 99.9%?
- Ablative analysis: Remove components from your system one at a time, to see how it breaks

Component	Accuracy
Overall system	99.9%
Spelling correction	99.0
Sender host features	98.9%
Email header features	98.9%
Email text parser features	95%
Javascript parser	94.5%
Features from images	94.0%

Conclusion: The email text parser features account for most of the improvement.

[baseline]

Slide Credit: Andrew Ng

# Design of ML Experiments: Summary

- Approach #1: Careful design
  - Spend a long term designing exactly the right features, collecting the right dataset, and designing the right algorithmic architecture
  - Implement it and hope it works
  - Benefit: Nicer, perhaps more scalable algorithms. May come up with new, elegant, learning algorithms; contribute to basic research in machine learning
- Approach #2: Build-and-fix
  - Implement something quick-and-dirty
  - Run error analyses and diagnostics to see what's wrong with it, and fix its errors
  - Benefit: Will often get your application problem working more quickly. Faster time to market

# Design of ML Experiments: Summary

- Time spent coming up with diagnostics for learning algorithms is time well-spent
- It's often up to your own ingenuity to come up with right diagnostics
- Error analyses and ablative analyses also give insight into the problem
- Two approaches to applying learning algorithms:
  1. Design very carefully, then implement
    - Risk of premature (statistical) optimization
  2. Build a quick-and-dirty prototype, diagnose, and fix