

# **CSE 6740 Lecture 1**

## ***What is Machine Learning?***

### ***(Overview)***

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

# Welcome to This Course

- CSE 6740 (88451, 88651) / ISYE 6740 (89018)
- Computational Data Analysis: Foundations of Machine Learning and Data Mining
- TuTh 1:35-2:55am, 2447 Klaus
- Office hours: Grab me right after a lecture
- `www.cc.gatech.edu/~agray/fall109.html`
- Mailing list: link from course webpage
- TA: Ravi Sastry `gmravi2003@gatech.edu`

# What This Course is

- This is GT's advanced machine learning course, though previous ML knowledge is not required.
- “All of machine learning” crash course – from 0 to 60 (beginner to professional) in one semester

# Goals of This Course

- Give you the foundations needed for:
  - Competent analysis of data (application of ML)
  - Design of new methods (ML research)
- Give you the big picture, including context for other courses
- Put forth a new version of that picture

# Taking this Course

- Yes, you can get into the class – if you can't register online, email me for a permit
- Yes, you should take this class, if you have the background
- Background: You will need basic calculus, basic linear algebra, basic probability

# Today

1. *What is machine learning?*
  - (a) Datasets
  - (b) Tasks of machine learning
  - (c) Parts of machine learning
  - (d) Relationship to other fields
2. *What is this course?*
  - (a) Overview of this course
  - (b) Main messages of the course
  - (c) Logistics

I'll stick around for questions.

# What is Machine Learning?

## Examples:

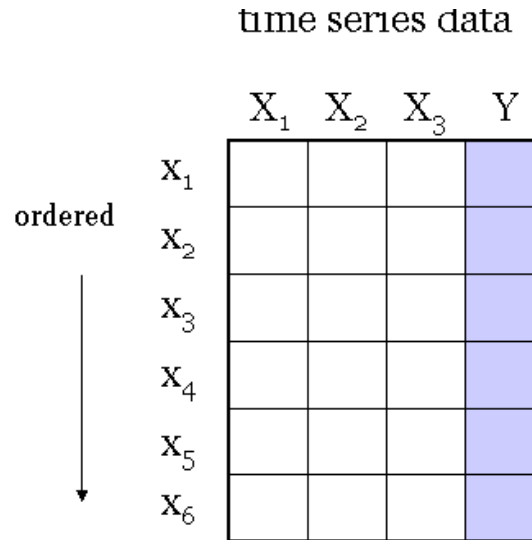
- Google: Targeted advertising
- Supermarkets: Promotion planning
- Call centers: Speech recognition
- Scanners: Optical character recognition
- Post office: Zipcode handwriting recognition
- Credit cards: Loan default prediction
- Stock market: Statistical arbitrage
- Drug design: Drug candidate screening
- Large Hadron Collider: Particle screening

# A Dataset

	$X_1$	$X_2$	$X_3$	$Y$
$x_1$				
$x_2$				
$x_3$				
$x_4$				
$x_5$				
$x_6$				

- *Features/attributes/dimensions*: columns
- *Data/points/instances/examples/samples*: rows
- *Target/outcome/response/label/dependent variable*: special feature to be predicted
- *Independent variables/covariates/predictors/regressors*: the other features

# Types of Data



- iid (independent identically distributed) vectors
- Time series (dependent vectors)
- Images (matrices)
- Variable-size non-vector data (e.g. strings, trees, graphs, text)
- Objects (e.g. within a relational schema)

# Main Goal of Learning: Prediction

The setup:

1. You obtain some kind of model of some *training data*, through a process called *learning* (also *estimation*).
2. Then you use that model to *predict* something about data you haven't seen before, but that comes from the same distribution as the training data, called *test data*.

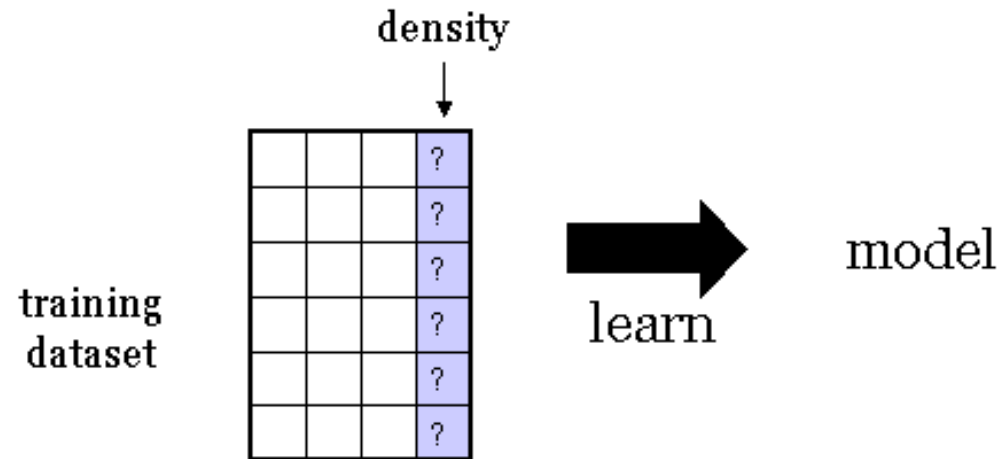
# 3 Main Learning Tasks

- Density estimation: predict the density
- Regression: predict a continuous target variable
- Classification: predict a discrete target variable

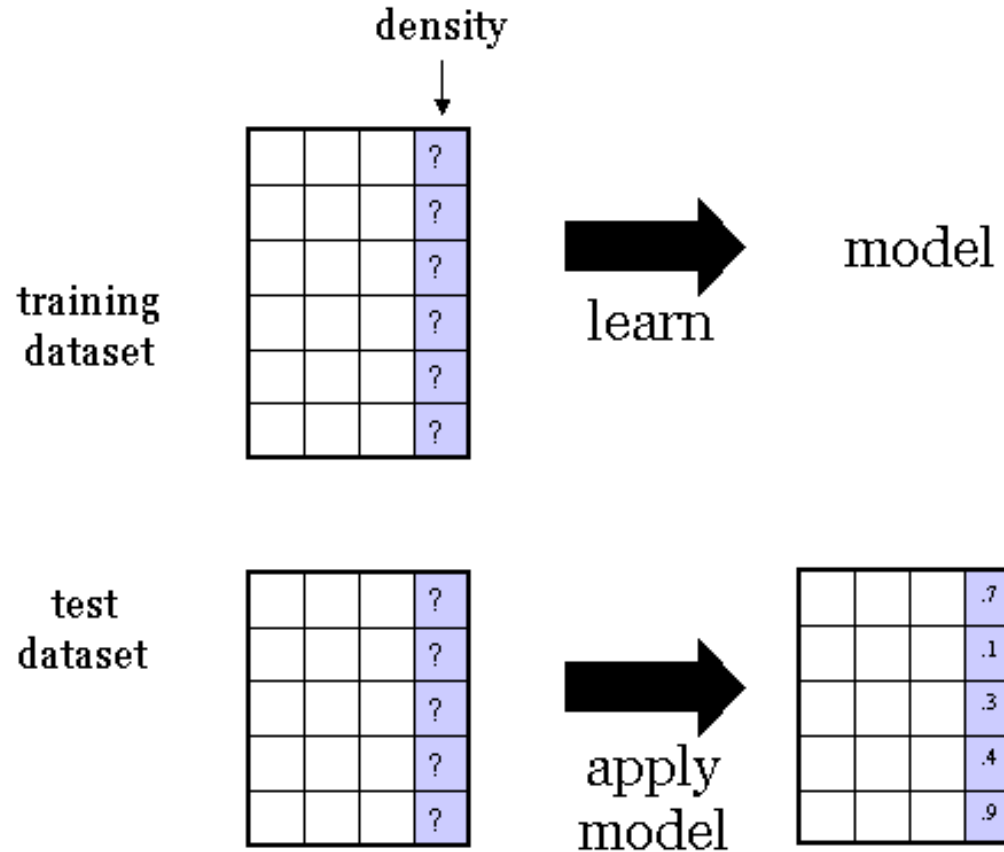
Others: clustering, dimensionality reduction.

- *Supervised learning*: We're predicting a target variable for which we get to see examples. (regression, classification)
- *Unsupervised learning*: We're predicting a target variable for which we never get to see examples. (density estimation, clustering, dimensionality reduction)

# Density Estimation



# Density Estimation



We never see the true value of the target. Unsupervised.

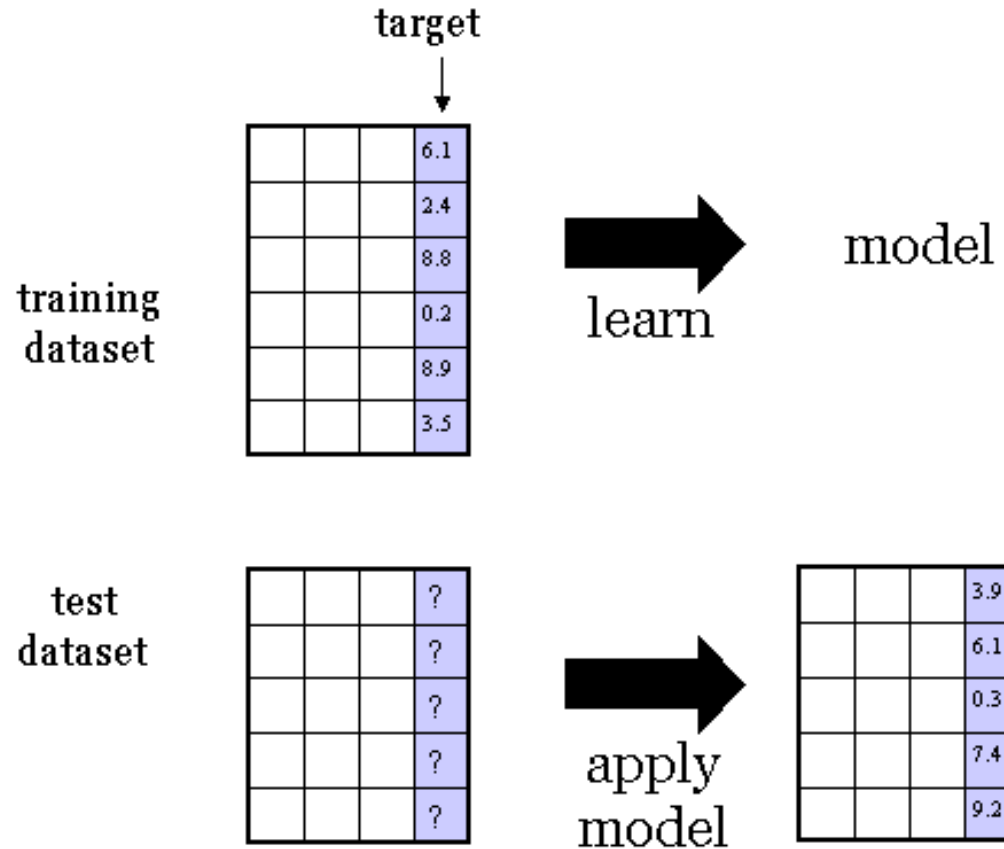
# Density Estimation

Example: observed matter in the sky

Should I fit a Gaussian? What would the right parameters be? Why?

Would a histogram be better? What would the right bin width be? Is it better than a Gaussian in general? Or worse?

# Regression



We're predicting a continuous target variable. Supervised.

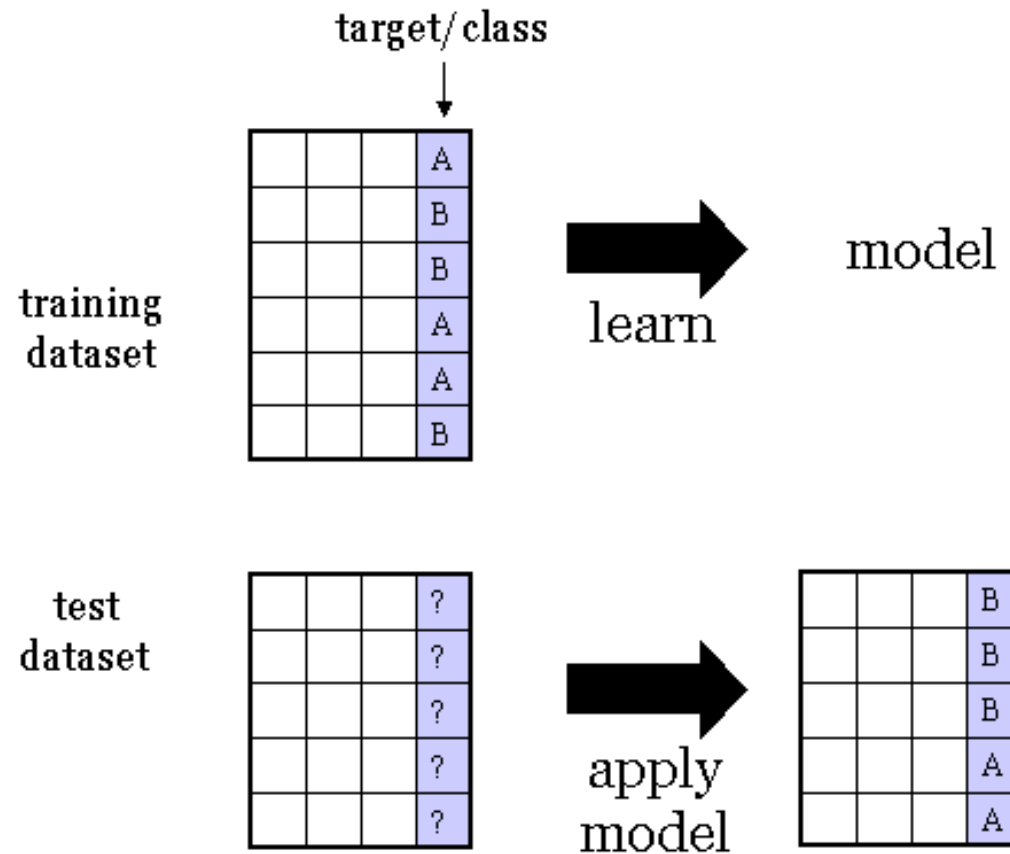
# Regression

Example: stock price prediction

Should I just predict the value of the last observation? How about a combination of the last  $k$  values (linear regression)? What's more general than that?

What should I expect my maximum error to be?

# Classification



We're predicting a discrete target variable. Supervised.

# Classification

Example: automatic zipcode digit recognition

Should I model each digit's images with a Gaussian (naive Bayes)? How about taking the class label of the nearest  $k$  training points ( $k$ -nearest-neighbor classifier)? Where does this method come from? How about if we focus on finding the widest decision boundary (support vector machine)? Where does this method come from? For this dataset, how can I definitely know one method is better than another?

How do these methods scale with dimensionality, statistically and computationally? Number of data? How do we estimate our future error accurately?

# Main Parts of Machine Learning

- Model class
- Loss (error) function
- Generalization mechanism
- Optimizer

Also sometimes important: evaluation algorithm.

# Model Classes

First we must pick a *model class*, or function class  $\mathcal{F}$ .

Parametric example (class of all Gaussians):

$$(1) \quad \mathcal{F} = \left\{ f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\}, \mu \in \mathbb{R}, \sigma > 0 \right\}$$

# Model Classes

Class of all parametric models in general:

$$(2) \quad \mathcal{F} = \{f(x; \theta) : \theta \in \Theta\}$$

*Nonparametric* example (class of all functions with a certain smoothness):

$$(3) \quad \mathcal{F} = \left\{ f : \int (f''(x))^2 dx < \infty \right\}$$

# Parameters

A *model* is an instance of the model class corresponding to a particular setting of the *parameters*  $\theta$ . (Confusingly, sometimes when we say “model” we mean the model class.)

$\theta^*$  = true, or best parameter value

$\hat{\theta}$  = estimated parameter value

$\hat{f}(X) = f(X; \hat{\theta})$  = estimated function

# Loss (Error) Function

Define a loss (error) function. Regression example:

$$(4) \quad L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$$

Classification example:

$$(5) \quad L(Y, \hat{f}(X)) = I(Y \neq \hat{f}(X))$$

Density estimation example:

$$(6) \quad L(Y, \hat{f}(X)) = -2 \log \Pr_{\hat{f}(X)}(Y)$$

# Learning and Prediction

*Generalization/test/prediction* error: The expected error on a new test data point:

$$(7) \quad E = \mathbb{E} \left[ L(Y, \hat{f}(X)) \right]$$

- *Learning/estimation/training/design*: try to find  $\hat{\theta}$  such that  $E$  is minimized – requires an *optimizer* and a *generalization mechanism*
- *Prediction/testing*: apply  $\hat{f}$  to predict  $Y$  for a new test set

Note that both of these are done on a computer. May be significant computations, sometimes requiring an efficient algorithm just to *evaluate* the models.

# Some Questions

- Which notion of error should we use? (loss functions)
- How do we ensure that the error on *future* data is minimized? (generalization)
- Which model/method should we use for our data? (model selection, hypothesis testing)
- What will the error of our method be, on future data? (error estimation, confidence band, learning theory)
- Are there methods that are optimal, under various assumptions? (asymptotic statistics)
- What will our method do when its assumptions don't hold? (robustness)

# What is Machine Learning?

Answer (logically speaking):

- *Statistics*  $\approx$  the science of inference from data
- *Machine learning*  $\approx$  multivariate statistics + computational statistics
- *Multivariate statistics*  $\approx$  prediction of values of a function assumed to underlie a multivariate dataset
- *Computational statistics*  $\approx$  computational methods for statistical problems (aka *statistical computation*) + statistical methods which happen to be computationally intensive
- *Data Mining*  $\approx$  exploratory data analysis, particularly with massive/complex datasets

# Inference

The process of using data to infer the distribution (or some aspect of it) that generated the data.

Main types of inference problems:

- Point estimation
- Confidence sets
- Hypothesis testing

Machine learning is mostly about point estimation.

# What is Machine Learning?

Answer (culturally speaking):

- *Statistics*: theory of inference, asymptotics, not just point estimation
- *Machine learning*: within point estimation: more emphasis on classification, implicitly nonparametric and computational
- *Data mining*: practical, interpretation and discovery, application-oriented, sometimes naive

You can ask me about the main conferences and journals in each of these areas.

# History of Machine Learning

- Multivariate statistics
- Pattern recognition, statistical (EE) - classification, high-dimensional (vision, speech), information theory
- Pattern recognition, syntactic (EE) - non-vector data
- AI (CS) - decision trees
- Cognitive scientists - neural nets
- Physicists - statistical physics, dynamical systems analogies
- CS theorists - learning theory

# History of Machine Learning

Lately:

- Return to parametric statistics and AI - graphical models; graph computations
- Return to pattern recognition - kernel machines; convex optimization computations
- Return to asymptotic statistics - ensemble methods
- Return to multivariate statistics - manifolds, kriging; linear algebra computations
- (I hope) Return to nonparametric statistics and EE - estimation theory; physics-based and geometric computations

# Growth of Machine Learning

Last 10 years:

- Applications in industry - data mining
- Applications in science - computational biology
- Fast-growing presence in AI, statistics, applied math

Reasons:

- Data is everywhere. This phenomenon is growing.
- Many modeling problems are more easily cast as data problems.
- Both widely useful and intellectually rich (mathematics, computation).

# Review of Syllabus

Basic concepts of ML, illustrated by 12 ML methods

- 2 **How do I learn a simple Gaussian?** Probability, random variables, distributions; estimation, convergence and asymptotics, confidence intervals
- 3 **How do I learn a mixture of Gaussians (MoG)?** Likelihood, the EM algorithm for MoG (i); generalization, model selection, cross-validation; k-means (ii), hidden Markov model (HMM) (iii)
- 4 **How do I learn any density?** Parametric vs. nonparametric estimation, Sobolev and other spaces; L2 error, kernel density estimation (KDE) (iv), optimal kernels, KDE theory

# Review of Syllabus

Basic concepts of ML, illustrated by 12 ML methods

- 5 How do I predict a continuous variable (regression)?** Linear regression (v), regularization, ridge regression and LASSO (vi); local linear regression (vii); conditional density estimation
- 6 How do I predict a discrete variable (classification)?** Bayes classifier, naive Bayes (ix), generative vs. discriminative; perceptron (x), weight decay, linear support vector machine (SVM) (xi); nearest-neighbor classifier (xii) and theory

# Review of Syllabus

General theory and model frameworks of ML

- 7 **Which loss function should I use?** Maximum likelihood estimation theory; L2 estimation, L2 MoG; Bayesian estimation, Bayesian MoG; minimax and decision theory, Bayesianism vs. frequentism
- 8 **Which model should I use?** AIC and BIC, Vapnik-Chernonenkis theory; cross-validation theory, the bootstrap

# Review of Syllabus

General theory and model frameworks of ML

- 9 **How can I learn fancier (combined) models?** Bagging, stacking, boosting; sieve theory
- 10 **How can I learn fancier (nonlinear) models?** Generalized linear models, logistic regression; Kolmogorov's theorem, generalized additive models; kernelization, reproducing kernel Hilbert spaces, nonlinear SVM, Gaussian process regression
- 11 **How can I learn fancier (compositional) models?** Recursive models, decision trees, hierarchical clustering; neural networks, backpropagation, deep belief networks; graphical models, mixtures of HMM's, conditional random field, max-margin Markov network; log-linear models; grammars

# Review of Syllabus

Further common ML problems and solutions

- 12 How do I reduce or relate the features?** Feature selection vs. dimensionality reduction, wrapper methods for feature selection; causality vs. correlation, partial correlation, Bayes net structure learning
- 13 How do I create new features?** principal component analysis (PCA), ICA, multidimensional scaling, manifold learning, supervised dimensionality reduction, metric learning
- 14 How do I reduce or relate the data?** Clustering, bi-clustering, constrained clustering; association rules and market basket analysis; ranking/ordinal regression; link analysis; relational data

# Review of Syllabus

Further common ML problems and solutions

- 15 How do I treat time series?** ARMA, Kalman filters and state-space models, particle filters; functional data analysis; change-point detection; cross-validation for time series
- 16 How do I treat non-ideal data?** Covariate shift; class imbalance; missing data, irregularly-sampled data, measurement errors; anomaly detection, robustness

# Review of Syllabus

General computational frameworks for ML

- 17 How do I optimize the parameters?** Unconstrained vs. constrained/convex optimization, derivative-free methods, first- and second-order methods, backfitting; natural gradient; bound optimization and EM
- 18 How do I optimize linear functions?** Computational linear algebra, matrix inversion for regression, singular value decomposition (SVD) for dimensionality reduction
- 19 How do I optimize with constraints?** Convexity, Lagrange multipliers, the KKT conditions, interior point method, SMO algorithm for SVM's

# Review of Syllabus

General computational frameworks for ML

- 20 How do I evaluate deeply-nested sums?** Exact graphical model inference, variational bounds on sums, approximate graphical model inference, expectation propagation
- 21 How do I evaluate large sums and searches?** Generalized N-body problems (GNP's), hierarchical data structures, nearest-neighbor search, fast multipole methods; Monte Carlo integration, Markov Chain Monte Carlo, Monte Carlo SVD
- 22 How do I treat even larger problems?** Parallel/distributed EM, parallel/distributed GNP's; stochastic gradient descent, online learning

# Review of Syllabus

Real-world application of ML

**23 How do I apply all this in the real world?** Overview of the parts of ML, choosing between the methods to use for each task, prior knowledge and assumptions; exploratory data analysis and information visualization; evaluation and interpretation, using confidence intervals and hypothesis tests, ROC curves; where the research problems in ML are

# Books

Required:

- *All of Statistics*, Wasserman
- *The Elements of Statistical Learning*, Hastie, Tibshirani, and Friedman
- *Pattern Recognition and Machine Learning*, Bishop

# Hidden Messages of This Course

I will emphasize the distinctions between:

- logical and cultural
- statistics and computation
- principles and methods
- theoretical and practical

I will also:

- blur cultural lines
- avoid current trends and dogma
- focus on theory that affects practice

# Grading

- 20% assignments: 3 assignments, involving some conceptual, mathematical, and programming questions based on C++ Fastlib/MLPACK (write pseudocode based on real code, write small code snippets)
- 30% final: on entire class, short-answer and multiple-choice
- 50% implementations/experiments: add (implement and test) one new ML method to MLPACK, from papers/books; starts after lecture 8 or so

# How Hard Will This Course Be?

- Pace: Fast. But hopefully clear.
- Mathematical, but no proofs to be written.
- Lots of implementation/experimentation work; roughly equivalent to writing one paper.

# Main Things You Should Know

- Main goal of ML
- Tasks of ML
- Parts of ML
- Whether this course is for you
- If you're taking it, expectations of the course

# Now to Hear From You...

- Any questions?
- Any special requests for advanced topics?