Introduction to Machine Learning

Umaa Rebbapragada, Ph.D.
Machine Learning and Instrument Autonomy Group

Astroinformatics 2019
Monday, June 24, 2019
California Institute of Technology

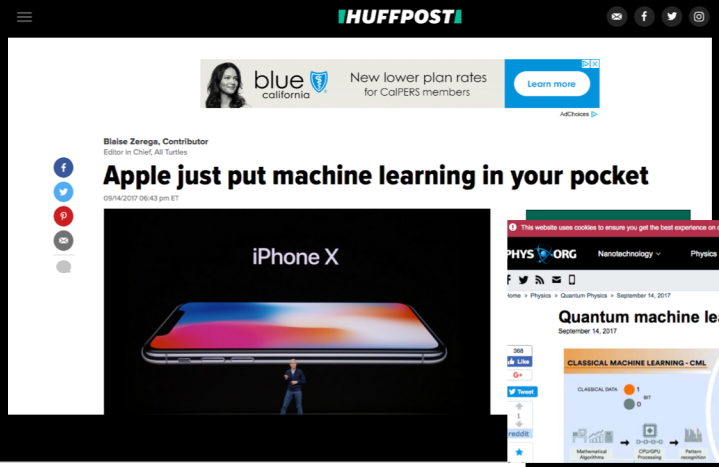
Research described in this presentation was carried out at the Jet Propulsion Laboratory under a Research and Technology Development Grant, under contract with the National Aeronautics and Space Administration.

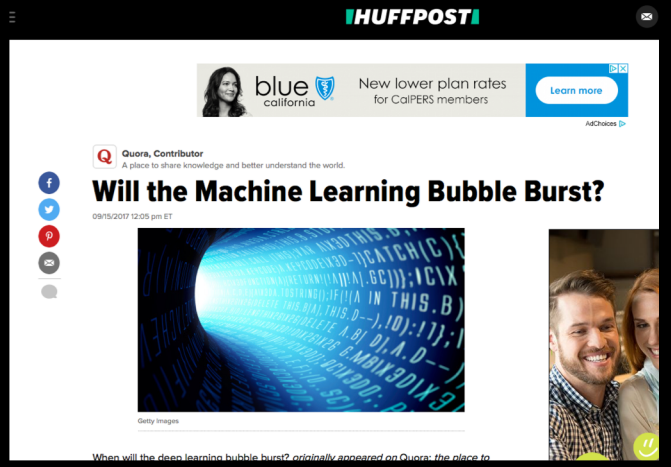
Jet Propulsion Laboratory
California Institute of Technology

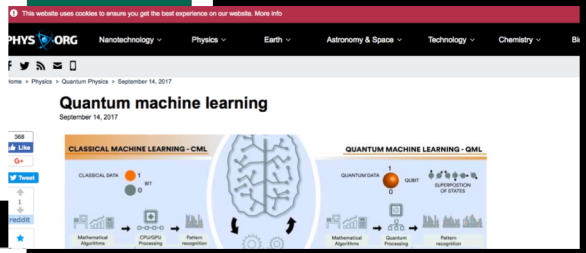
What is Machine Learning

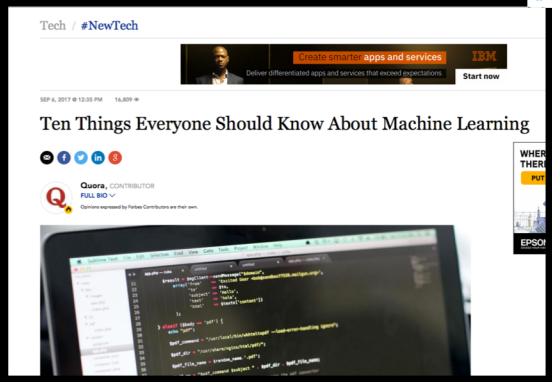
Machine Learning

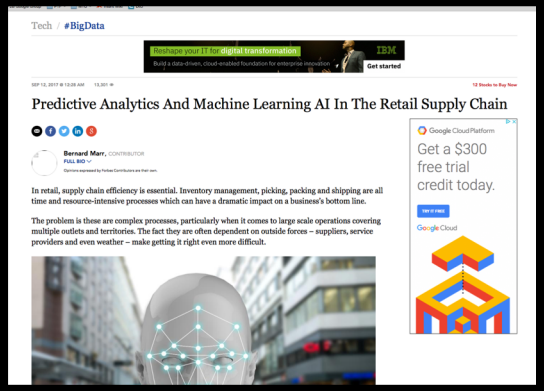
Everyone's Talking About It











http://www.businessnewsdaily.com/10215-get-a-job-in-artificial-intelligence.html
http://www.huffingtonpost.com/entry/will-the-machine-learning-bubble-burst_us_59bb60fce4b0390a1564dc7b
http://www.huffingtonpost.com/entry/apple-just-put-machine-learning-in-yourpocket_us_59b9c368e4b06b71800c3694
https://phys.org/news/2017-09-quantum-machines.html
https://www.forbes.com/sites/quora/2017/09/06/ten-things-everyone-should-know-about-machine-learning/#2cc264ff4e9e
https://www.forbes.com/sites/bernardmarr/2017/09/12/predictive-analytics-and-machine-learning-ai-in-the-retail-supply-chain/#501091692c7d

What is Machine Learning?

- Initially a subfield of artificial intelligence
- Computers “learn” from data/experience rather than explicitly coded rules
- Nexus of statistics, information theory, signal processing, optimization methods

Outline

- Supervised Learning
  - Ingredients
  - Overfitting and Other Key Concepts
  -
- Unsupervised Learning
  - Clustering
  - Anomaly Detection

- Training Set Triage

Supervised Learning

Data

Features

Examples

	# Pixels	Axis Length	Half Width	Median Flux	…
1	40	17.97	1.36	14.0	
2	49	16.77	2.00	13.0	
3	52	21.20	1.29	13.9	
4	92	32.42	0.86	24.2	
5	233	44.28	1.20	26.1	
6	61	13.25	1.37	170.3	
7	47	16.15	0.98	24.2	
8	120	25.71	1.01	119.7	
9	62	13.95	1.42	44.3	
10	180	29.09	1.35	19.9	
.					
.					
.					
N					

Data

for a classification task

Features

Class label

Examples

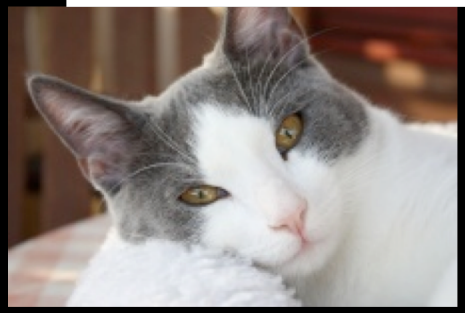
	# Pixels	Axis Length	Half Width	Median Flux	…	Real / Bogus
1	40	17.97	1.36	14.0		Bogus
2	49	16.77	2.00	13.0		Bogus
3	52	21.20	1.29	13.9		Bogus
4	92	32.42	0.86	24.2		Real
5	233	44.28	1.20	26.1		Real
6	61	13.25	1.37	170.3		Bogus
7	47	16.15	0.98	24.2		Bogus
8	120	25.71	1.01	119.7		Real
9	62	13.95	1.42	44.3		Bogus
10	180	29.09	1.35	19.9		Real
.						
.						
.						
N						

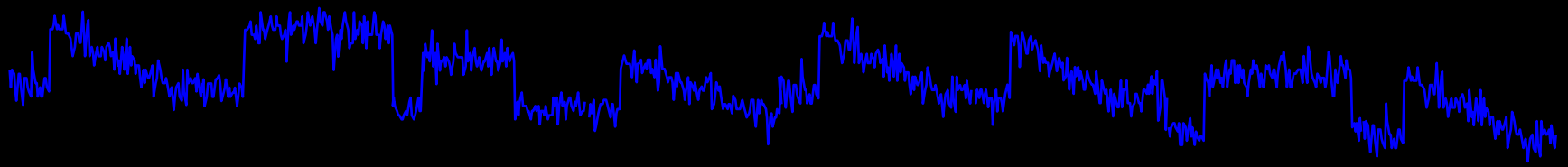
Representing Data

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed at turpis vitae velit euismod aliquet. Pellentesque et arcu. Nullam venenatis gravida orci. Pellentesque et arcu. Nam pharetra. Vestibulum viverra varius enim.

Nam laoreet dui sed magna. Nunc in turpis ac lacus eleifend sagittis. Pellentesque ac turpis. Aliquam justo lectus, iaculis a, auctor sed, congue in, nisl. Aenean luctus vulputate turpis. Mauris urna sem, suscipit vitae, dignissim id, ultrices sed, nunc.

Phasellus nisi metus, tempus sit amet, ultrices ac, porta nec, felis. Quisque malesuada nulla sed pede volutpat pulvinar. Sed non ipsum. Mauris et dolor. Pellentesque suscipit accumsan massa. In consectetuer, lorem eu lobortis egestas, velit odio



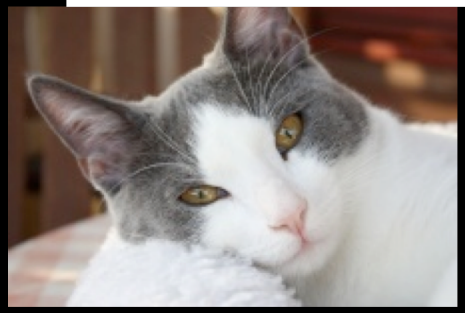


Representing Data

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed at turpis vitae velit euismod aliquet. Pellentesque et arcu. Nullam venenatis gravida orci. Pellentesque et arcu. Nam pharetra. Vestibulum viverra varius enim.

Nam laoreet dui sed magna. Nunc in turpis ac lacus eleifend sagittis. Pellentesque ac turpis. Aliquam justo lectus, iaculis a, auctor sed, congue in, nisl. Aenean luctus vulputate turpis. Mauris urna sem, suscipit vitae, dignissim id, ultrices sed, nunc.

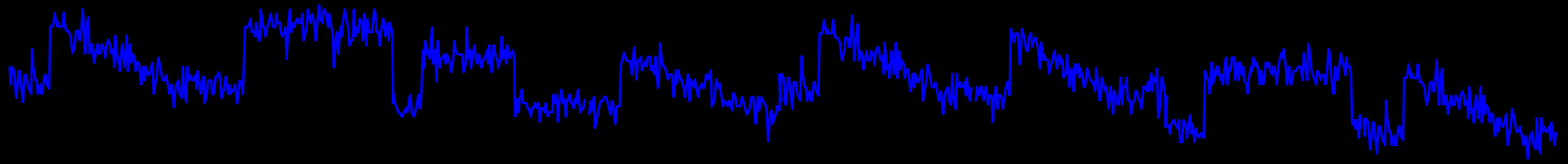
Phasellus nisi metus, tempus sit amet, ultrices ac, porta nec, felis. Quisque malesuada nulla sed pede volutpat pulvinar. Sed non ipsum. Mauris et dolor. Pellentesque suscipit accumsan massa. In consectetuer, lorem eu lobortis egestas, velit odio



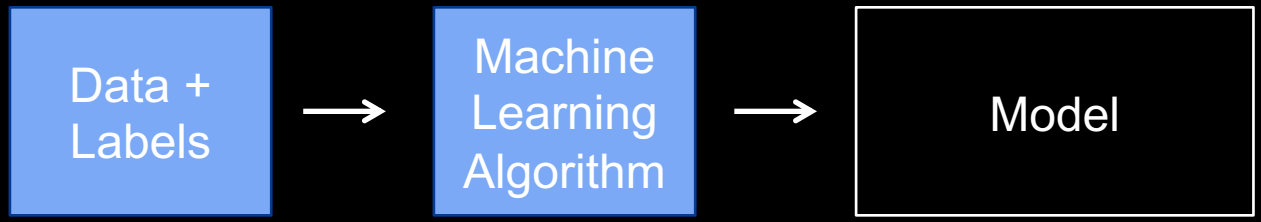
Bag of Words
TFIDF

Pixel values,
SIFT, HoG,
histograms of visual words

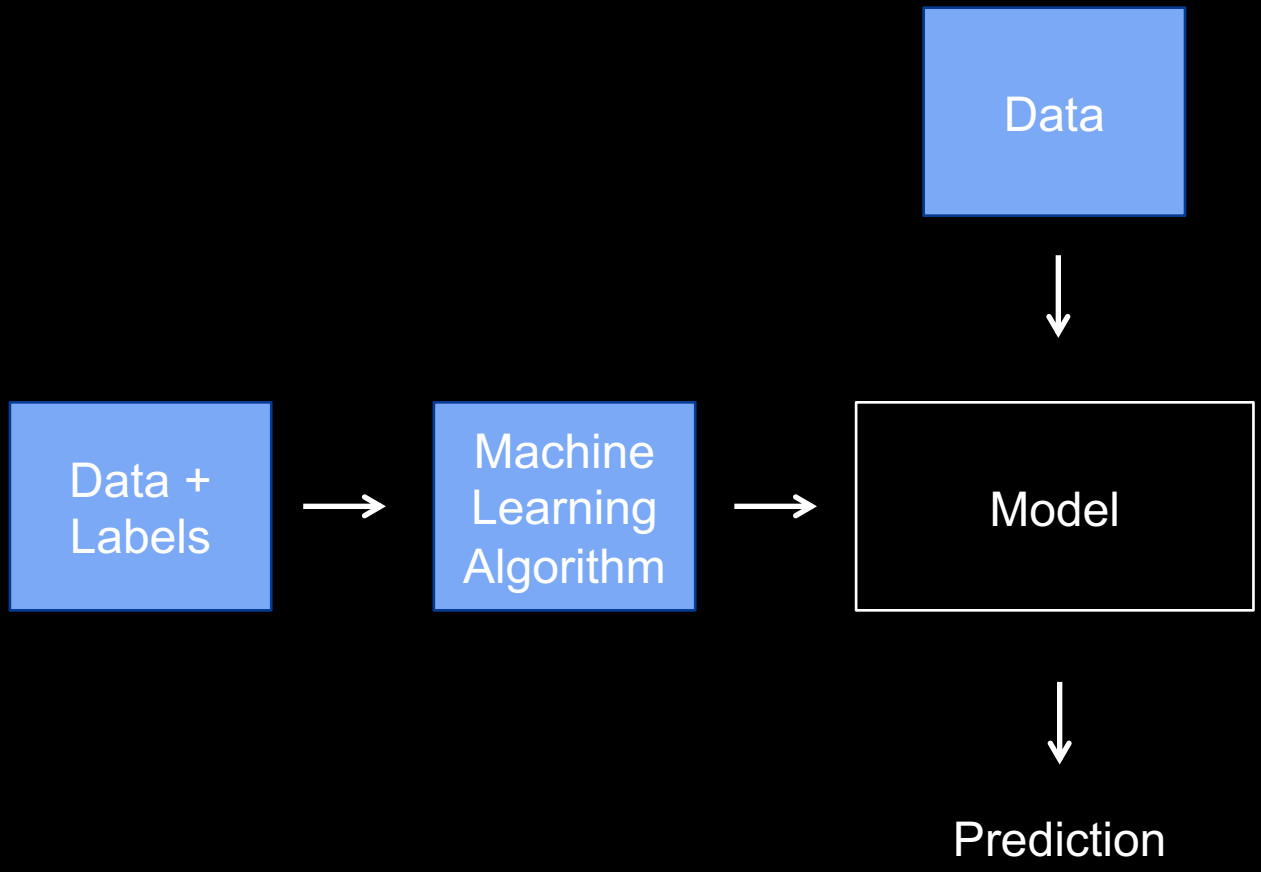
DFT, wavelets,
time series statistics



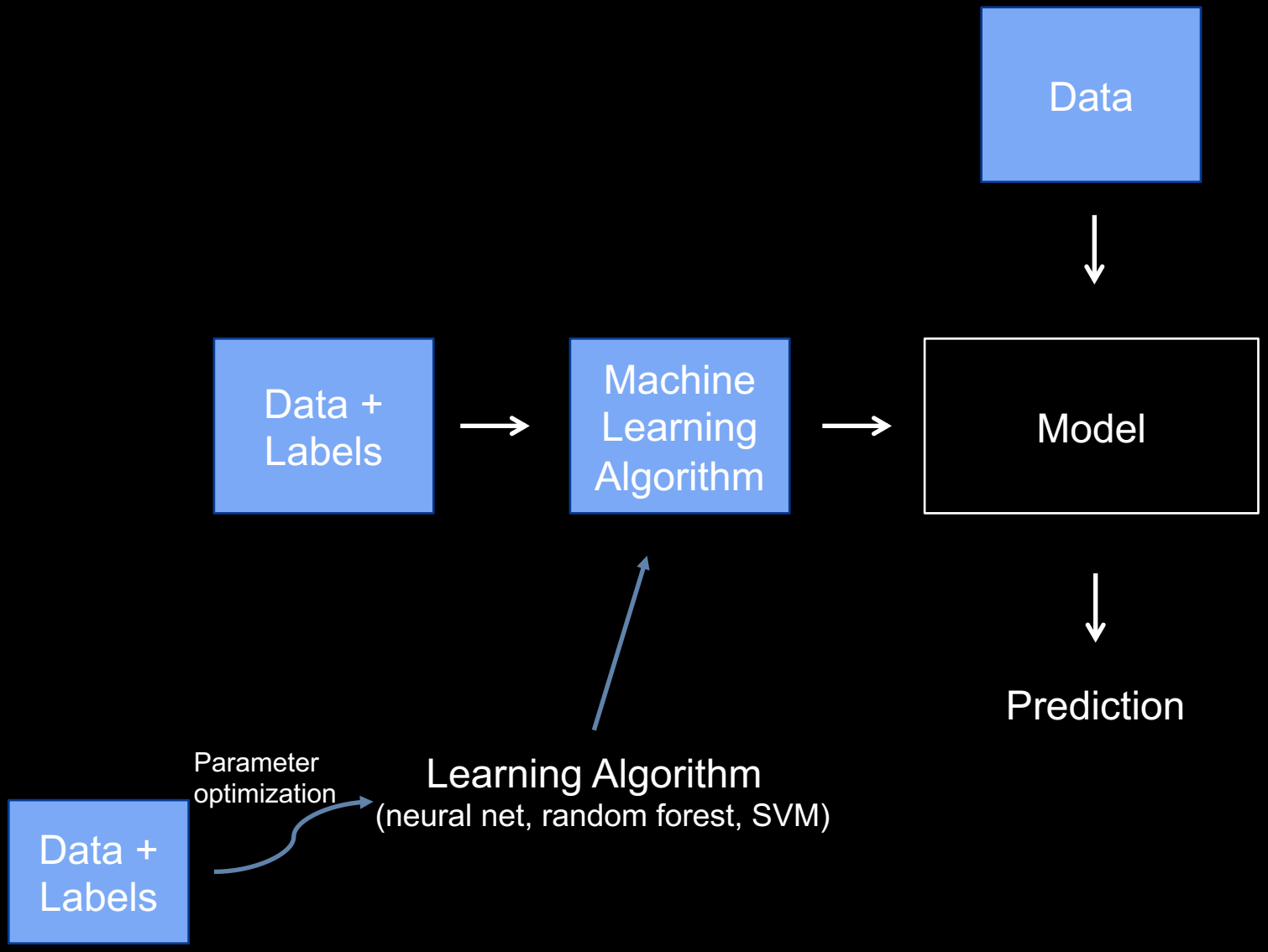
Training a Classifier



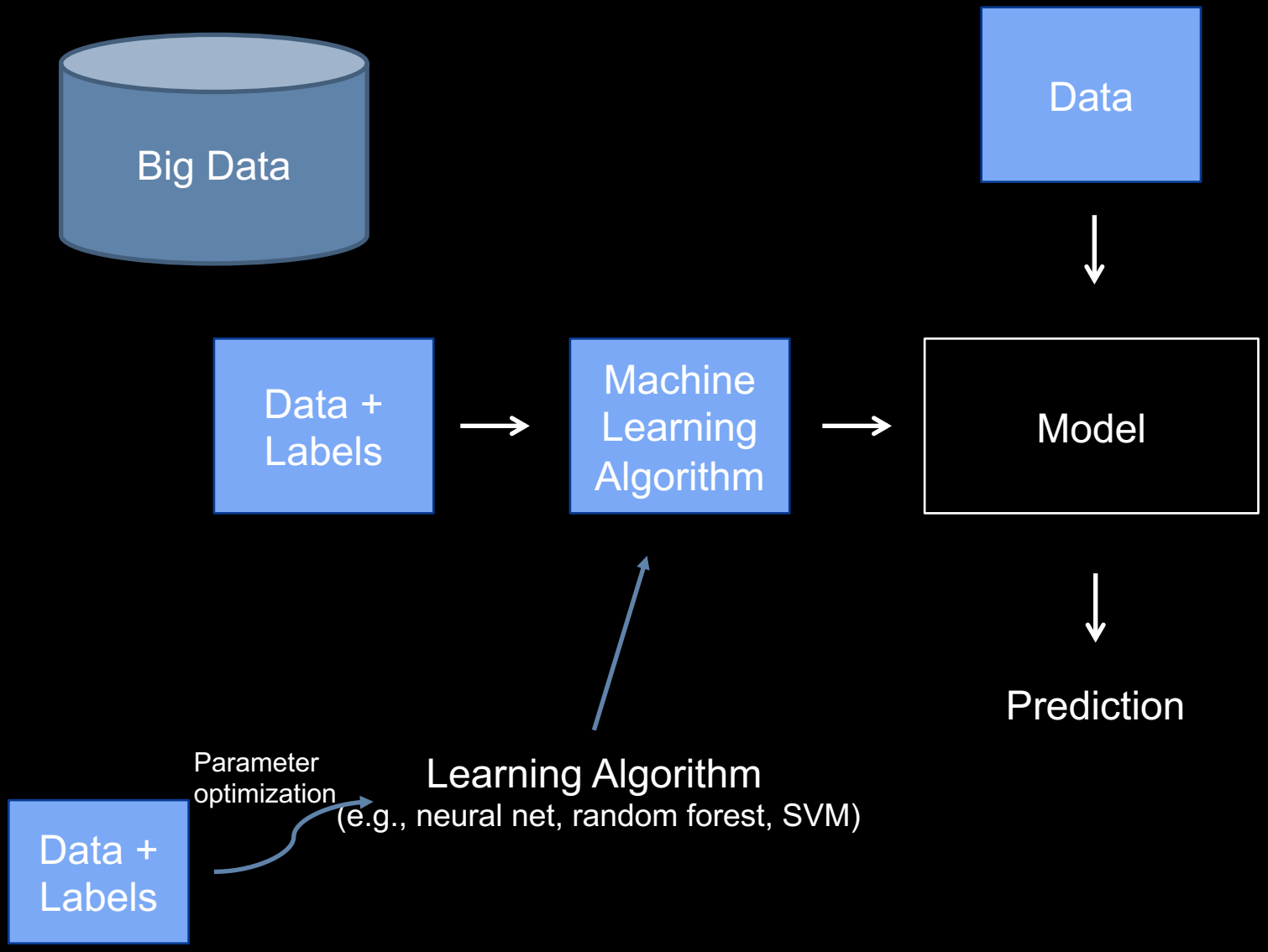
Training a Classifier



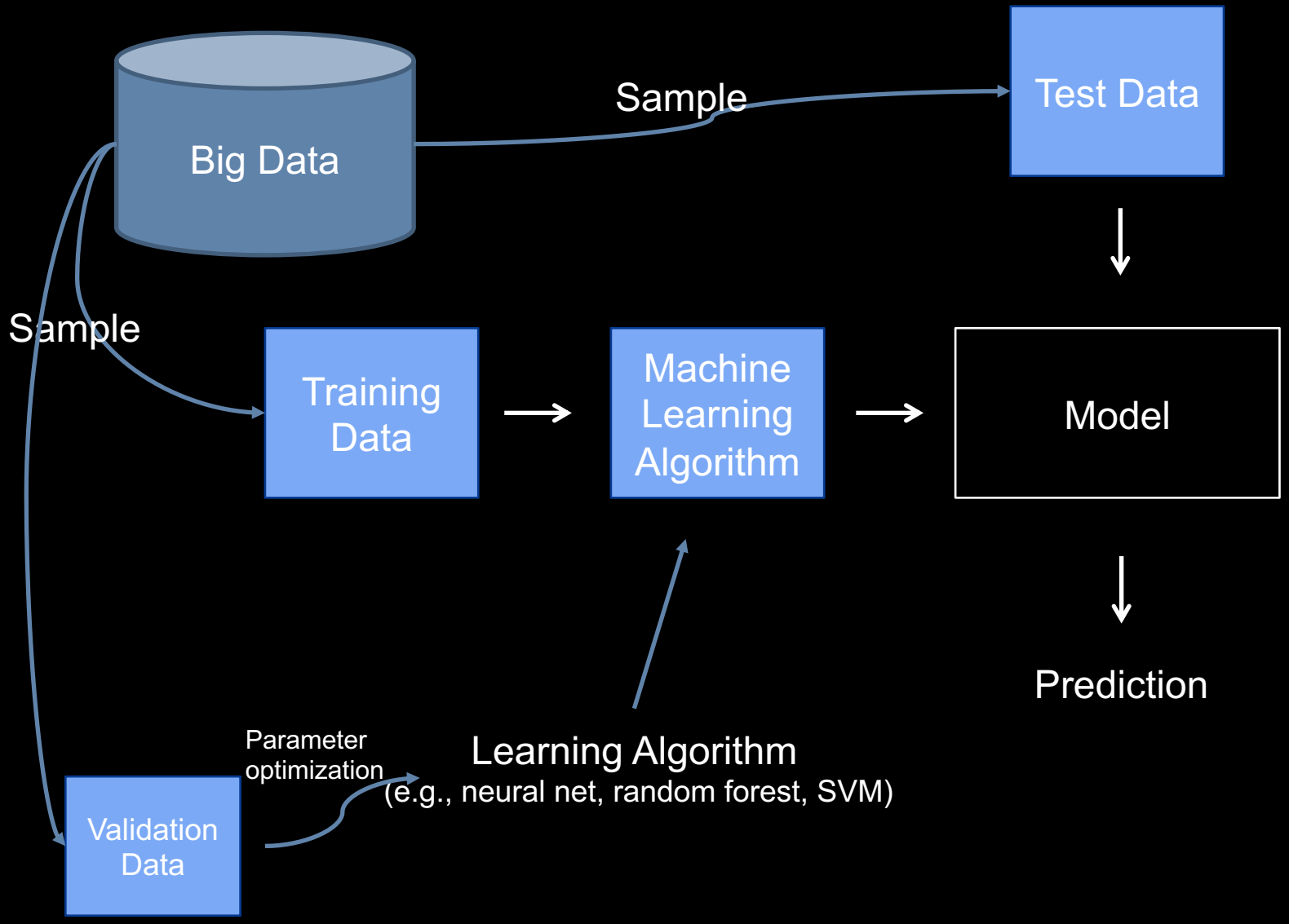
What are the Ingredients?



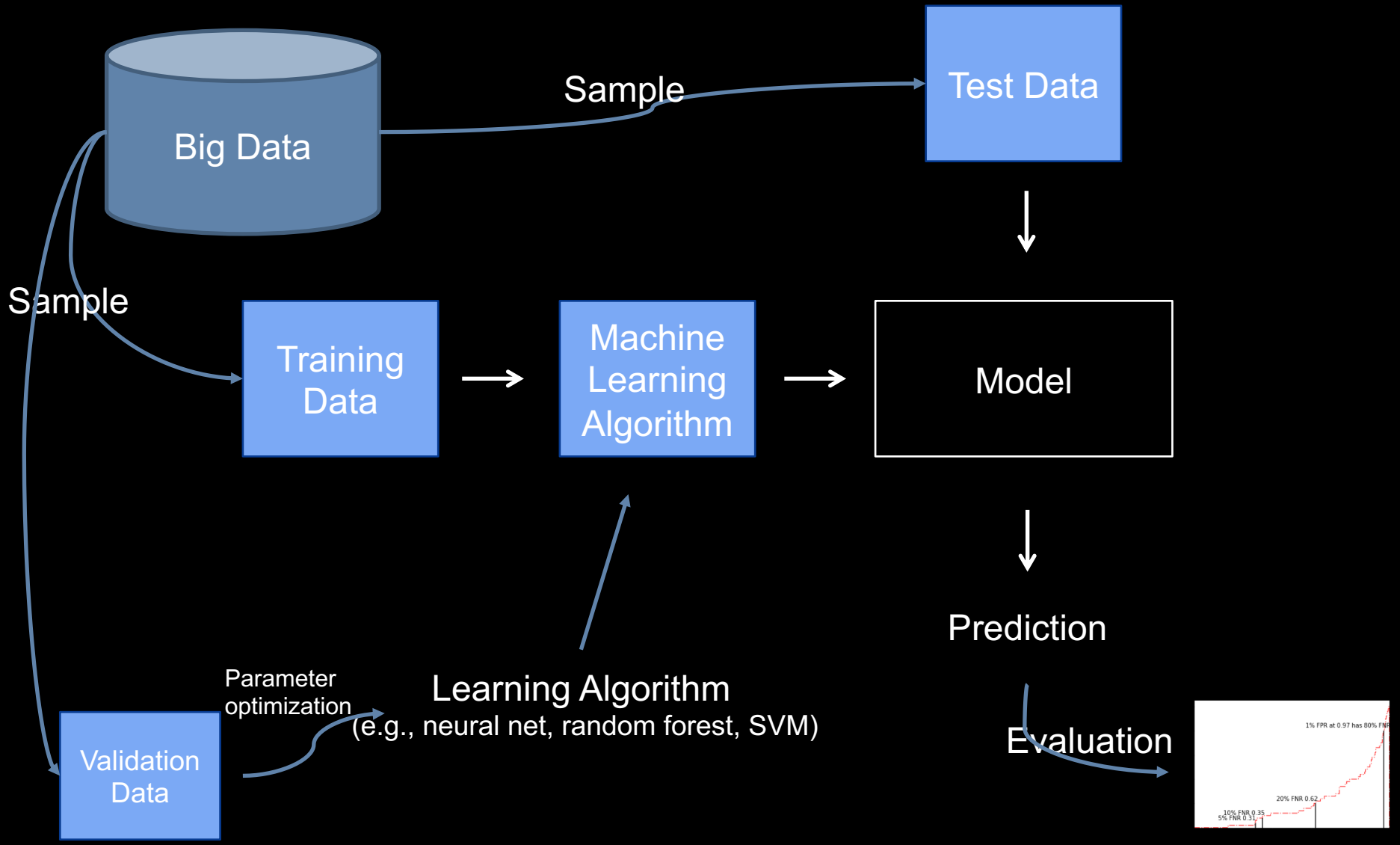
What are the Ingredients?



What are the Ingredients?



What are the Ingredients?



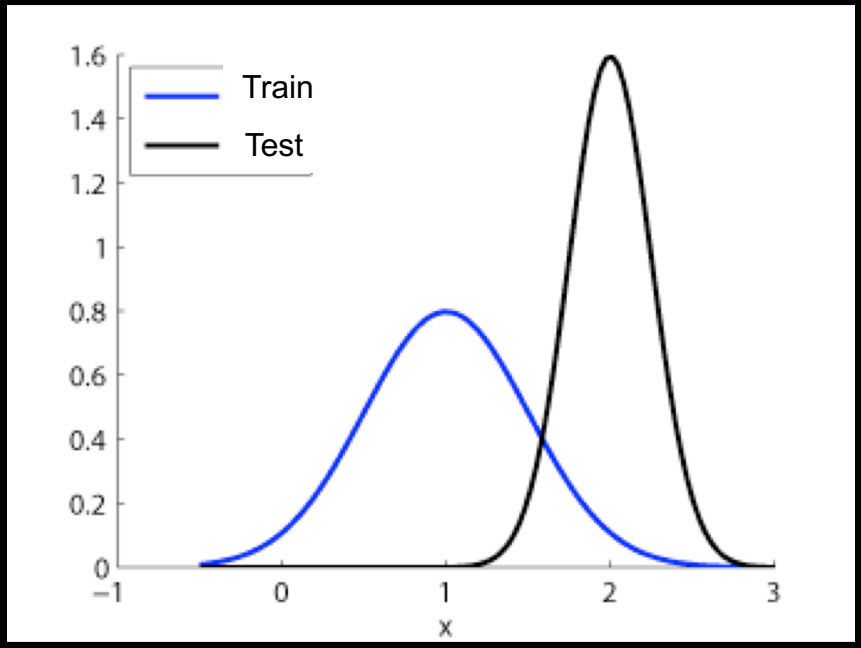
Ingredients Summarized

- Sampling Data into Training, Validation, and Test Sets
- Feature Representation
- Learning Algorithm
- Evaluation Metric

Ingredient: Sampling Data

Key Assumption

Train, validation, and test set examples should be sampled from the same data distribution



Source: http://www.ms.k.u-tokyo.ac.jp/software.html

Consider the Following Situations

- Wide-field time domain astronomical survey:
  - Can I train on data collected on extra-galactic fields, and apply to new data coming in from Galactic Plane
- Earthquake damage detection
  - Can I train on the earthquake in Christchurch, NZ, and apply to imagery from Haiti
- Clinical Trials:
  - Can I train on a patient population in Netherlands, and apply the model to patients in the USA?
- Different astronomical filters
  - Can I train on r-band and apply to g-band?

Train / Test / Validation Splits

- Conventional wisdom for small , medium datasets (up to 100K)
  - 70/30 Split for Train/Test
  - 60/20/20 Split for Train/Validation/Test
  - Cross validation is also an option
  - Grid Search within Cross Validation also an option
- Deep Learning era (1M and more)
  - 98/1/1 😳
- Test set should be large enough to give you high confidence on your application.
- Minority classes should be represented in your smaller sets.

How to Split your Labeled Data

- Consider a pixel classification problem using this RGB satellite image
- How would sklearn divide this image into a train and test set?

How to Split your Labeled Data

Labeled Data

Pixel #	R	G	B	Label
1				
2				
3				
4				
.				
.				
1M				

How to Split your Labeled Data

Pixel #	R	G	B	Label
1				
2				
3				
4				
.				
.				
1M				

Labeled Data

Training Data

Pixel #	R	G	B	Label
1				
2				
4				
5				
.				
.				
1M				

Test Data

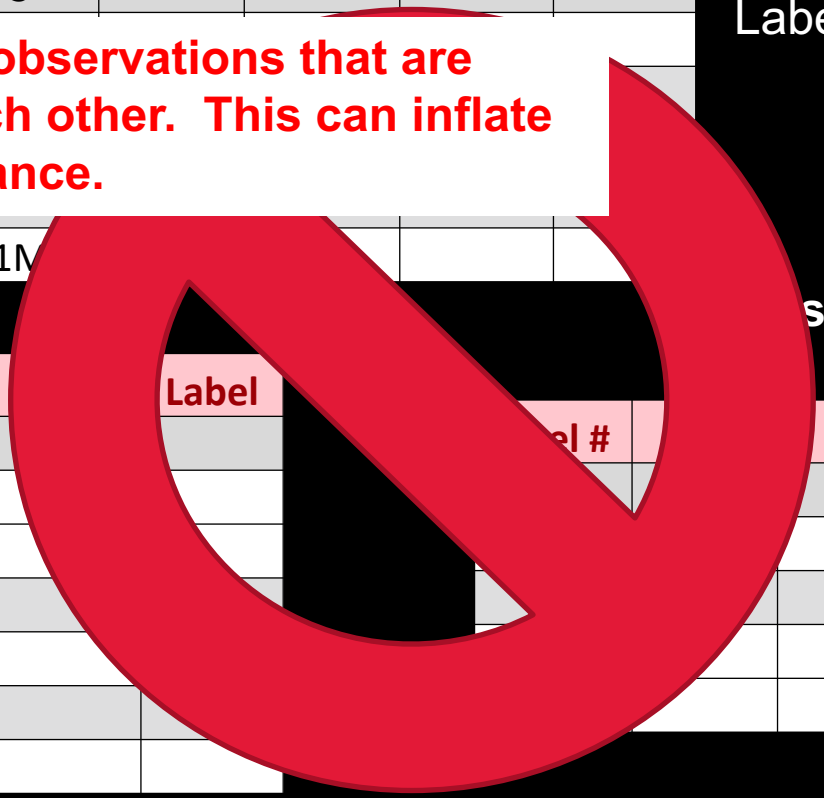
Pixel #	R	G	B	Label
3				
6				
.				
.				
1M				

How to Split your Labeled Data

Pixel #	R	G	B	Label
1				
2				
3				
1M				

Labeled Data

Do not split adjacent observations that are nearly identical to each other. This can inflate your test set performance.



Training Data

Pixel #	R	G		Label
1				
2				
4				
5				
.				
.				
1M				

st Data

el #		G	B	Label

How to Split your Test Data

- Can anyone think of an example in astronomy?

How to Split your Test Data

- Can anyone think of an example in astronomy?
- Example: ZTF takes two exposures within minutes of each other. If a transient isn't present in both, the source is rejected. However, if a transient is present, both candidates are getting saved.
- How can you protect against sklearn?

How to Split your Test Data

- Can anyone think of an example in astronomy?
- Example: ZTF takes two exposures within minutes of each other. If a transient isn't present in both, the source is rejected. However, if a transient is present, both candidates are getting saved.
- How can you protect against sklearn?
- Answer: you have to write your own cross validation splitting strategy. Fortunately, sklearn allows you to do this.

Getting Labels

- Experts annotate
- Amateurs via Crowdsourcing Platforms (e.g., Zooniverse)
- Ground Truth
- Cross-matching to Reliable Catalogs
- Which are the most reliable?

Getting Labels

Ranked

- Experts annotate
- Amateurs via Crowdsourcing Platforms (e.g., Zooniverse)
- Ground Truth
- Cross-matching to Reliable Catalogs
- Spectroscopy

- Don’t like to label negative examples
- Don’t know what they’re doing
- Robots can’t go everywhere
- Error Rate
- Not all objects can be followed up

Ingredient: Learning Algorithms

Types of Learning Algorithms

- Linear Models (logistic regression, perceptron)
- Instance-based learning (k-nearest neighbors)
- Neural nets (multi-layer perceptron, CNNs, RNNs, LSTMs)
- Decision trees
- Ensemble methods (Random forests, Bagging, Boosting)
- Support Vector Machines
- Bayesian Networks (Hidden Markov Models, Naïve Bayes)

Learning Algorithm Ingredients

- Learning = Representation + Evaluation + Optimization
- Representation: Classifier must be represented in a formal computing language. Represents all the possible sets of classifers, called a **hypothesis space**.
- Evaluation: scoring or objective function used during the learning process to distinguish between good and bad hypotheses. Will learn the classifier that minimizes error on the training set
- Optimization: Method for search the hypothesis space for the best classifiers.

Domingos, P. CACM 12 “A Few Useful Things to Know about Machine Learning”

Popular Algorithms Broken Down

Table 1: The three components of learning algorithms.

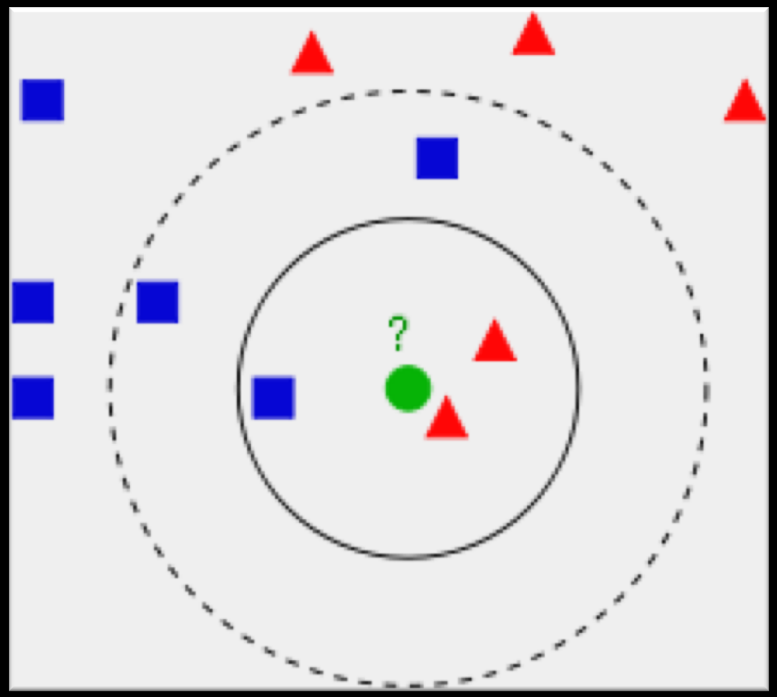
Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic 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		

Domingos, P. CACM 12 “A Few Useful Things to Know about Machine Learning”

Three Examples

Algorithm	Representation	Evaluation	Optimization
kNN			
Logistic Regression			
Decision Tree			

k-Nearest Neighbors (kNN)



- Training Data:
  - Blue squares
  - Red triangles
-  is a query point
- K = 3, classify as 
- K = 5, classify as 
- The majority vote of the closest K neighbors of the training set determines the predicted label

k-Nearest Neighbors (kNN)

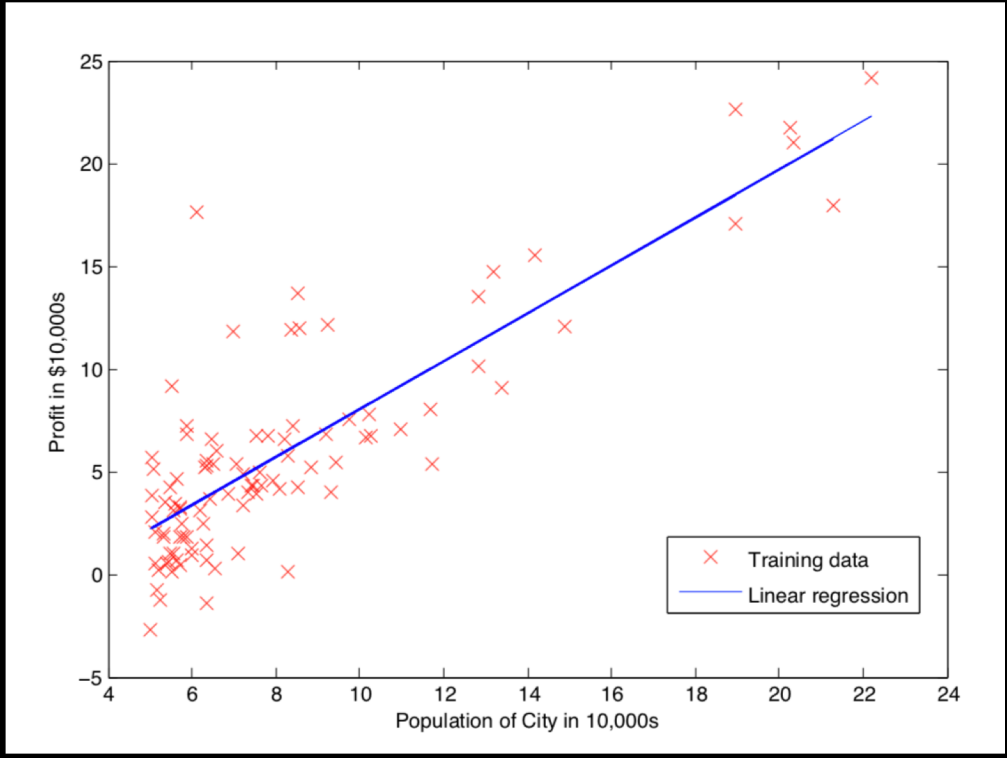
Algorithm	Representation	Evaluation	Optimization
kNN	Example	Squared Distance	Greedy Search
Logistic Regression			
Decision Tree			

Logistic Regression

- Recall linear regression is fitting a model in order to predict a continuous-valued output given input features. Because h is linear, the cost junction J is convex and has global minimum.

$$h_\theta(x) = \theta^T x$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$



Source: Andrew Ng, Introduction to Machine Learning, Coursera

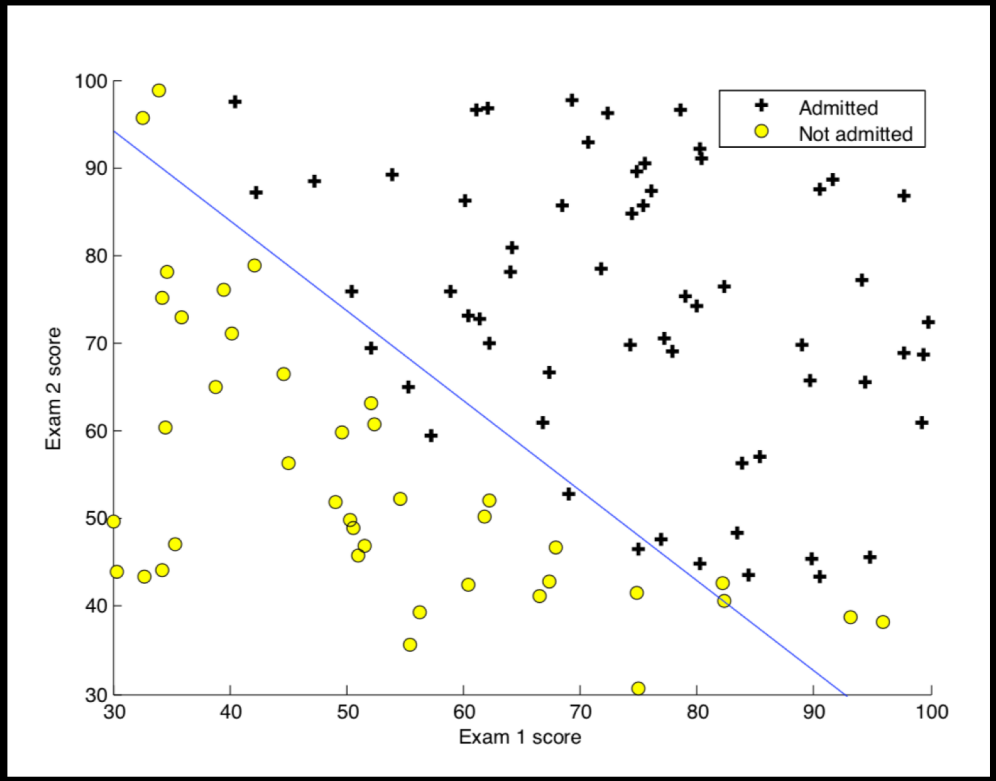
Logistic Regression

- Logistic regression hypothesis wraps the linear regression hypothesis in the logistic function to output a prediction scaled to [0,1]. The cost function is the same, but it's no longer convex.

$$h_\theta(x) = g(\theta^T x),$$

$$g(z) = \frac{1}{1 + e^{-z}}.$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$



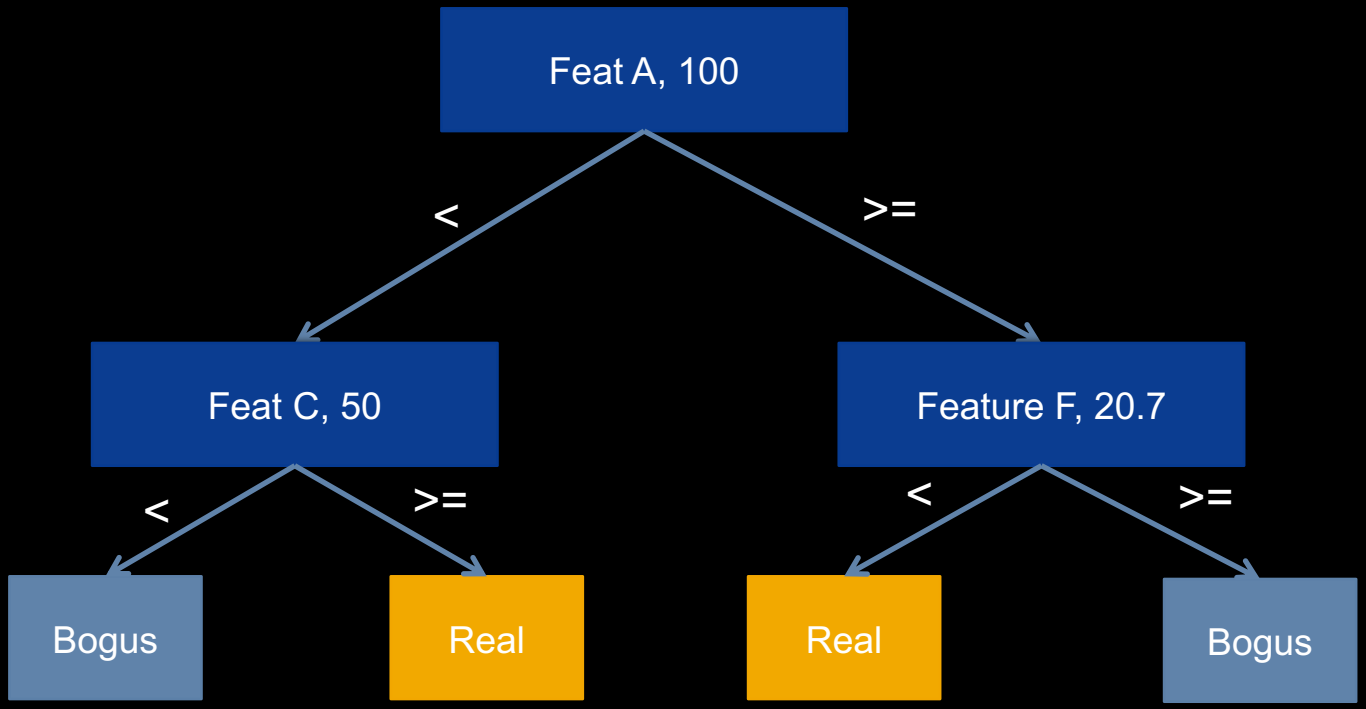
Source: Andrew Ng, Introduction to Machine Learning, Coursera

Logistic Regression

Algorithm	Representation	Evaluation	Optimization
kNN	Example	Squared Distance	Greedy Search
Logistic Regression	Hyperplane	Squared Error	Gradient Descent
Decision Tree			

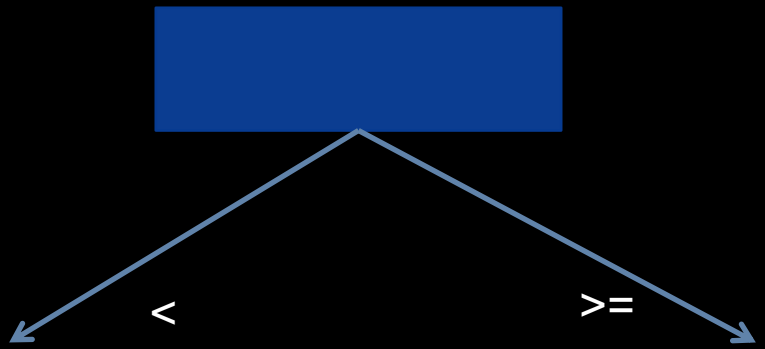
Decision Tree

- Example of a 3-node decision-tree built for a binary problem.
- Classification time is fast
- Concatenation of rules, easy for humans intuit



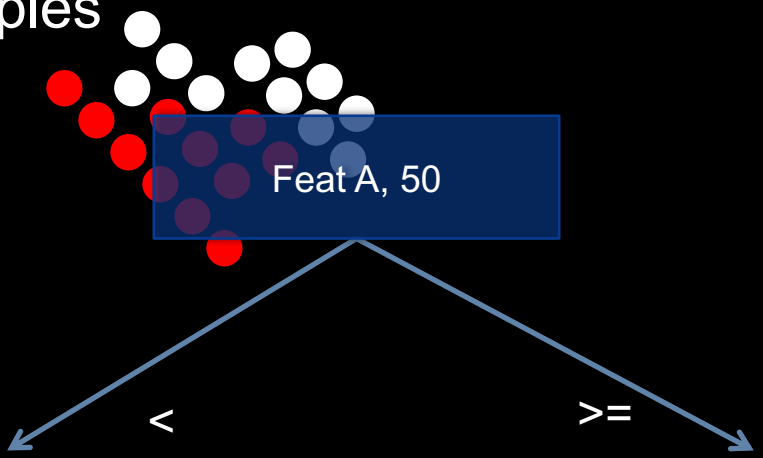
Training a Decision Tree

- How does the learning algorithm decide which feature and feature value to split on?



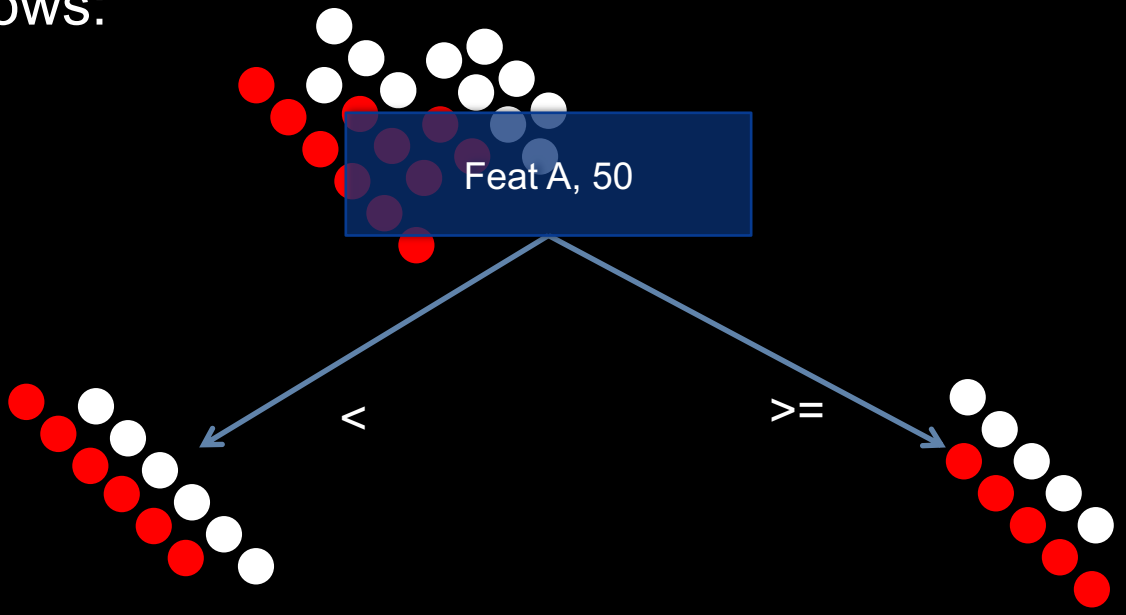
Training a Decision Tree

- Consider Feature A, and threshold value 50, and our set of training examples



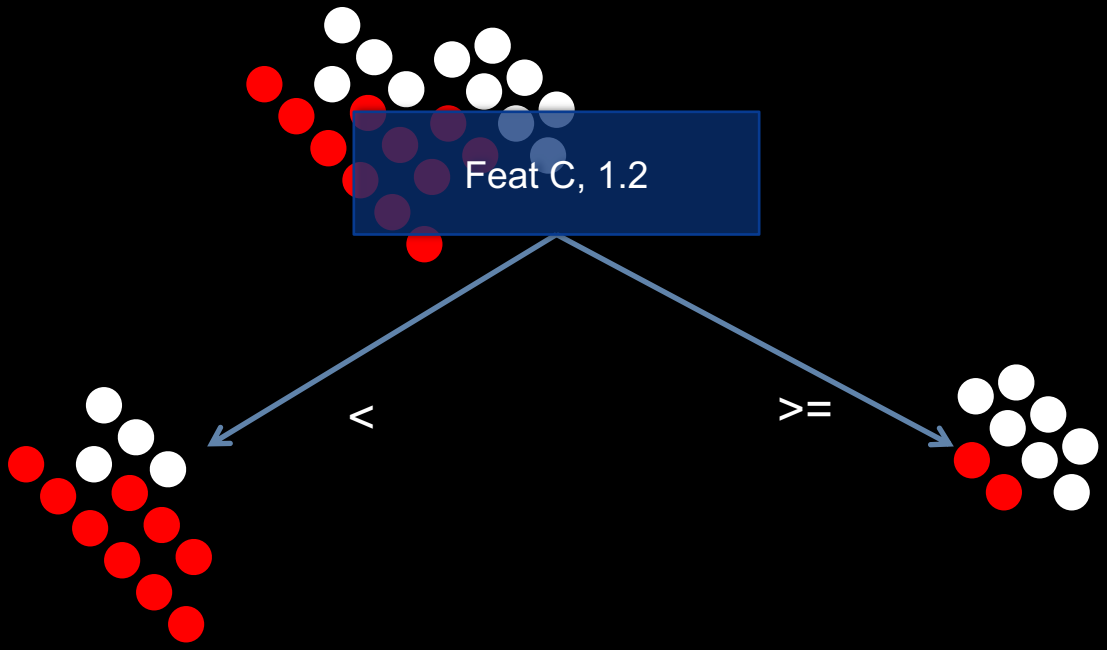
Training a Decision Tree

- This feature, feature value pair partitions my training samples as follows:



Training a Decision Tree

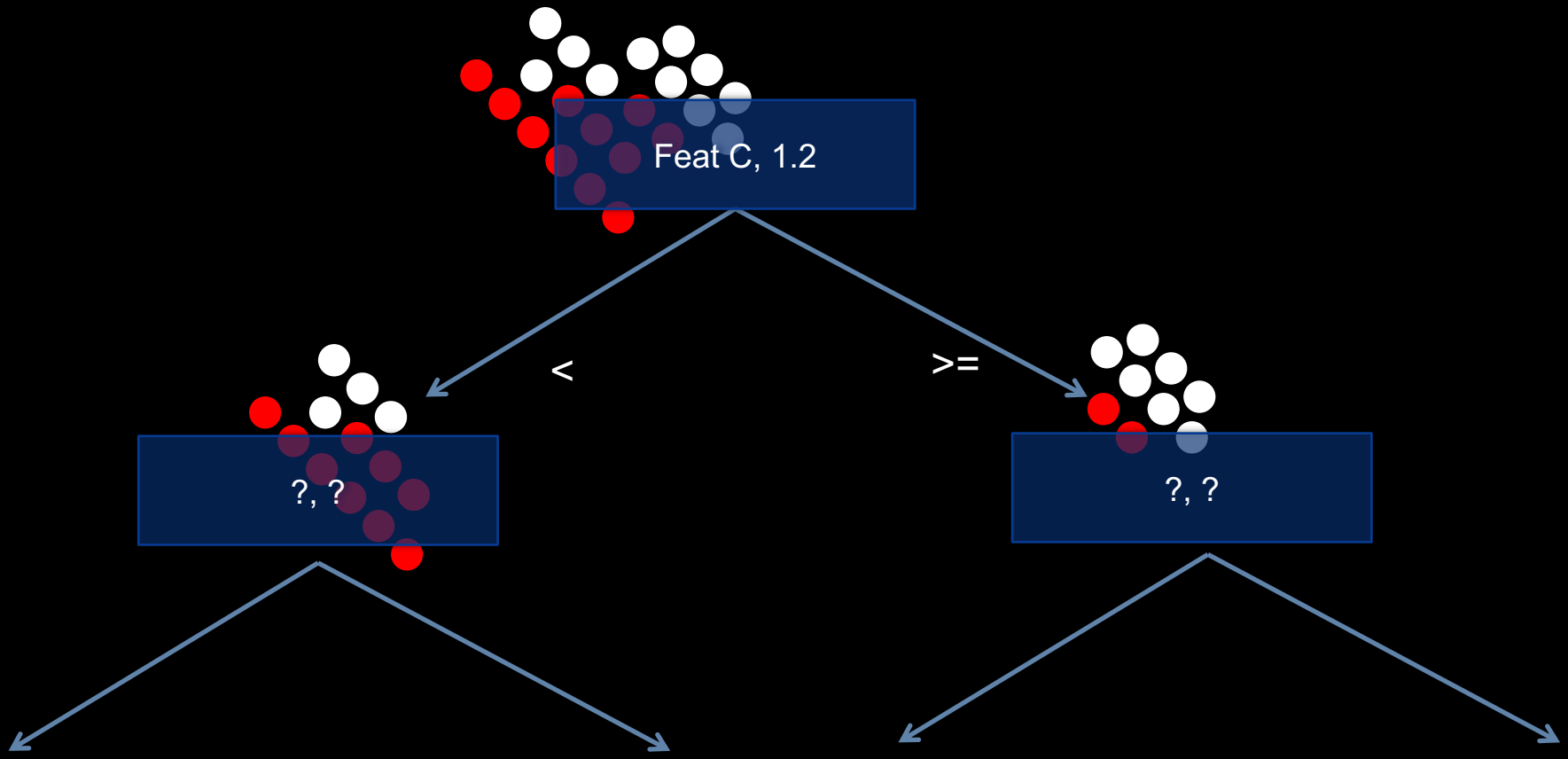
- This feature, feature value pair partitions my training samples as follows:



- Which split is preferable?

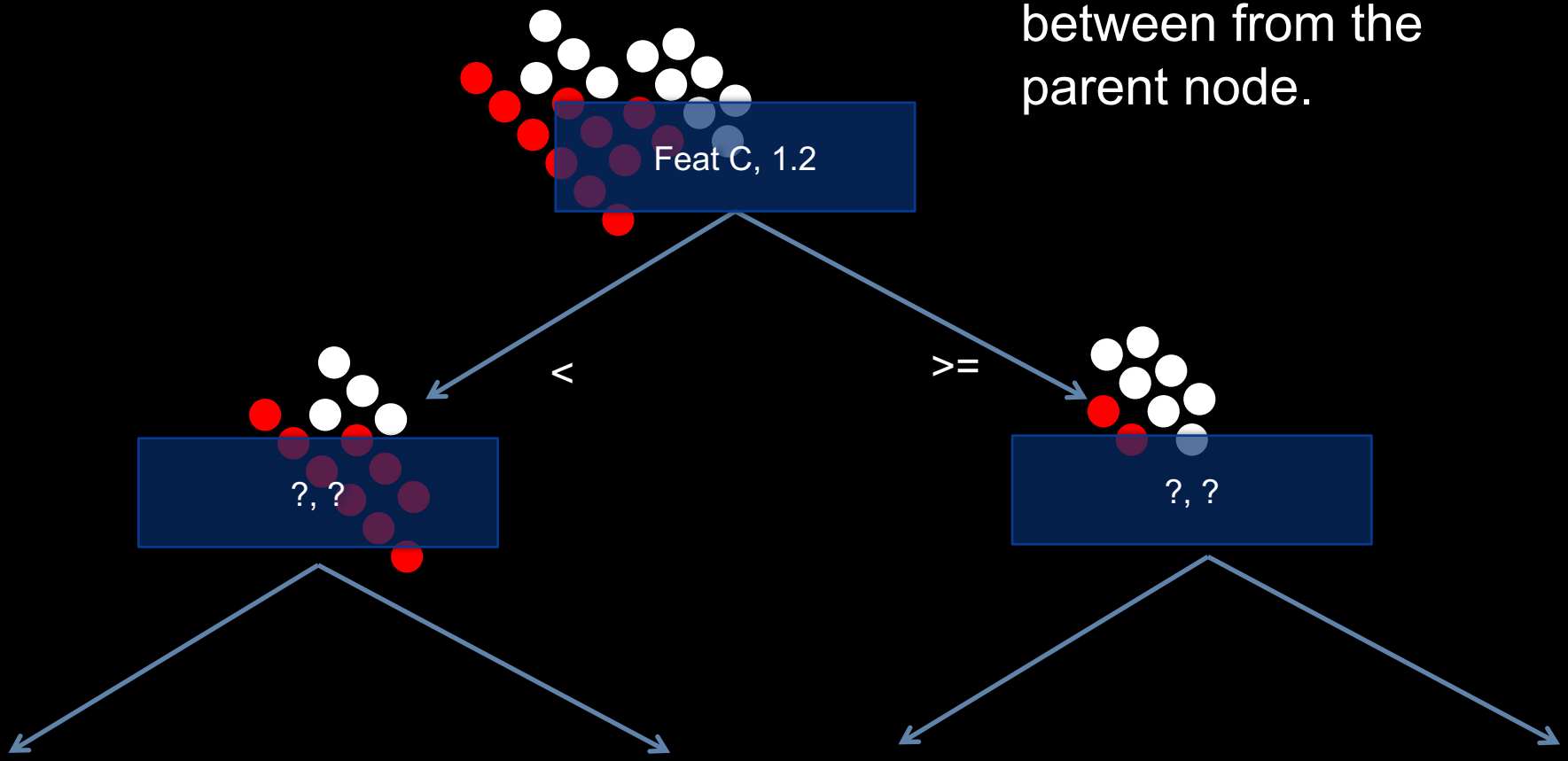
Training a Decision Tree

- We recursively continue this operation with the sub-samples at each child node until purity of classification is achieved



Training a Decision Tree

Label purity of the sub-samples at each node are calculated using Information Gain, which is a decrease in entropy between from the parent node.



Training a Decision Tree

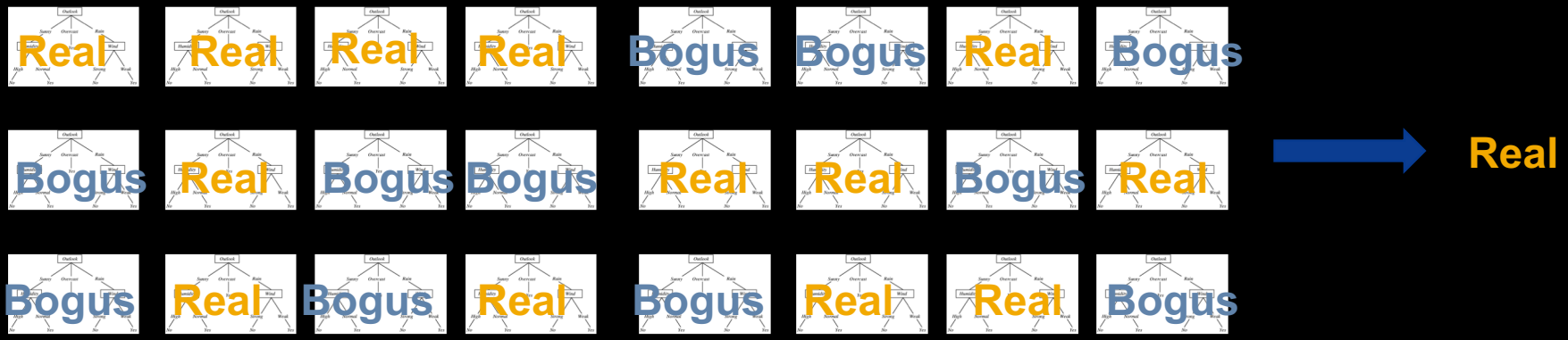
- Each node defines a unique feature sub-space, as opposed to logistic regression or kNN which is always operating in the complete feature space
- Decision trees can grow quite long.
- Usually only a random subset of (feature, feature value) pairs are considered at each node during training

Decision Tree

Algorithm	Representation	Evaluation	Optimization
kNN	Example	Squared Distance	Greedy Search
Logistic Regression	Hyperplanes	Likelihood	Gradient Descent
Decision Tree	Binary, K-ary Tree	Information Gain	Greedy Search

Random Forest and Ensemble Methods

- Build many models by repeatedly sampling data with replacement
- Vote on final classification
- Ensembles reduces generalization error of single tree models



Which One to Choose?

- Test Set Accuracy
  - Labeled data that's been held out for testing
- Training Time vs. Run Time
  - e.g., train on ground, run onboard
- Number of Parameters to tune
  - Computationally expensive to perform a grid search over full hyper-parameter space
- Scales in number of features, examples
- Word of mouth

Ingredient: Evaluation

How to Evaluate

- Independent Test Sets
  - obtain another set of test data
- Cross Validation
  - reserve portion of labeled data for testing, rotate that fold, average results

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Measuring Performance

- Confusion Matrix
- Accuracy = (TP + TN) / # examples

Predicted

Actual	Positive (1)	Negative (0)
Positive (1)	**True Positive (TP)**	**False Negative (FN)**
Negative (0)	**False Positive (FP)**	**True Negative (TN)**

Measuring Performance for Binary Problems

- False Positive Rate (FPR) = FP / (FP + TN)

Actual	Predicted: Positive (1)	Predicted: Negative (0)
Positive (1)	True Positive (TP)	False Negative (FN)
Negative (0)	**False Positive (FP)**	**True Negative (TN)**

- False Negative Rate (FNR) = FN / (TP + FN)

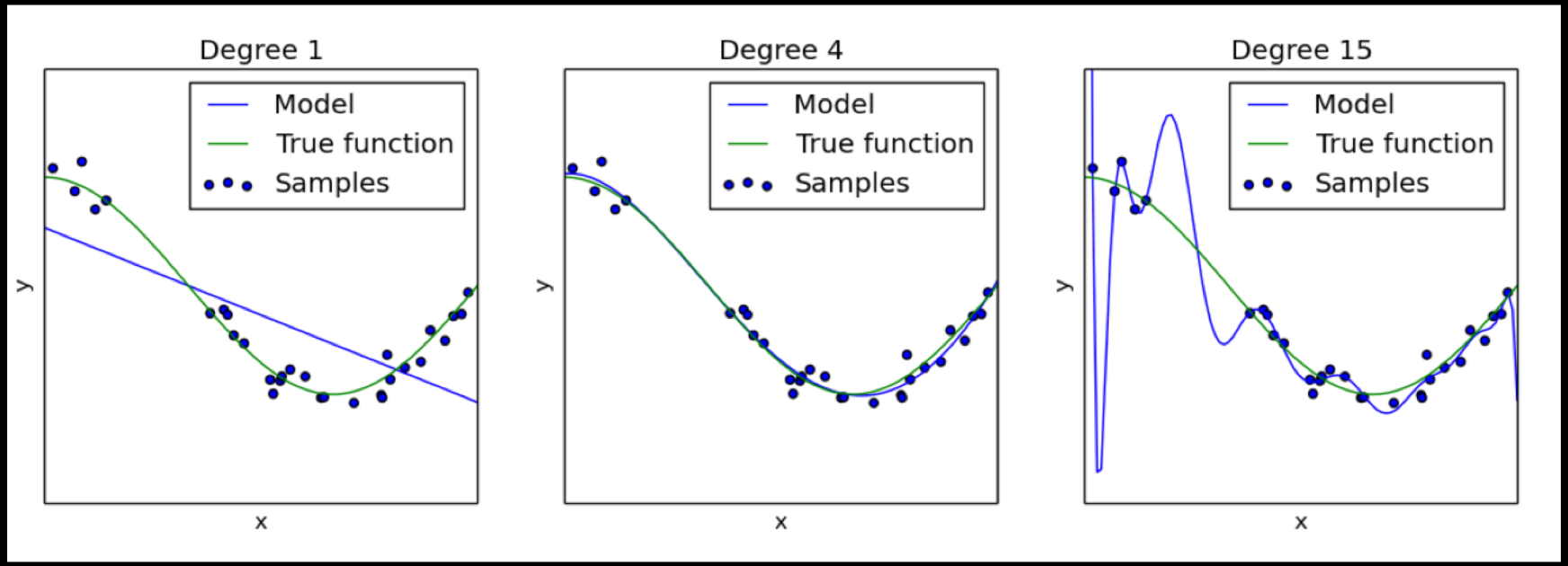
Actual	Predicted: Positive (1)	Predicted: Negative (0)
Positive (1)	**True Positive (TP)**	**False Negative (FN)**
Negative (0)	False Positive (FP)	True Negative (TN)

Overfitting and Other Key Concepts

Goal: Generalization

- Goal: build a model that generalizes well on test examples
- Training set error is the error associated with the model fit on your training data.
- Test set error is the error associated with the model fit on your test data.
- Oftentimes, training error is much better than test error.
- **A classifier that generalizes well should have a low test error.**
- A classifier that has a low training error but a high test error is said to be **overfit**.

Underfitting vs. Overfitting

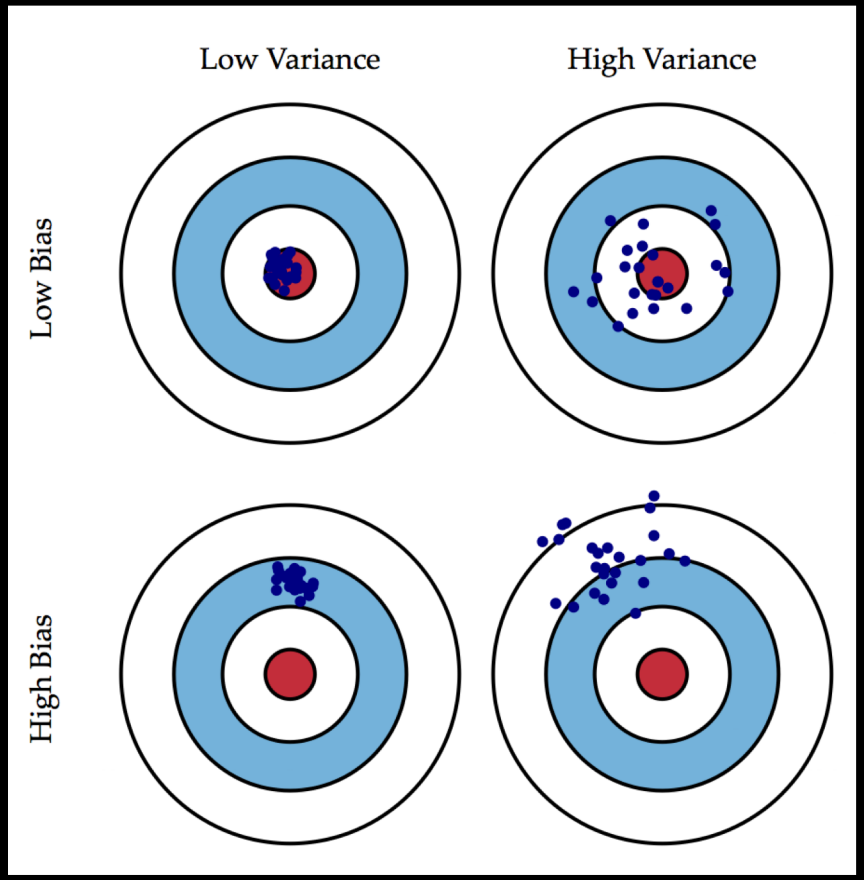


Underfitting

Overfitting

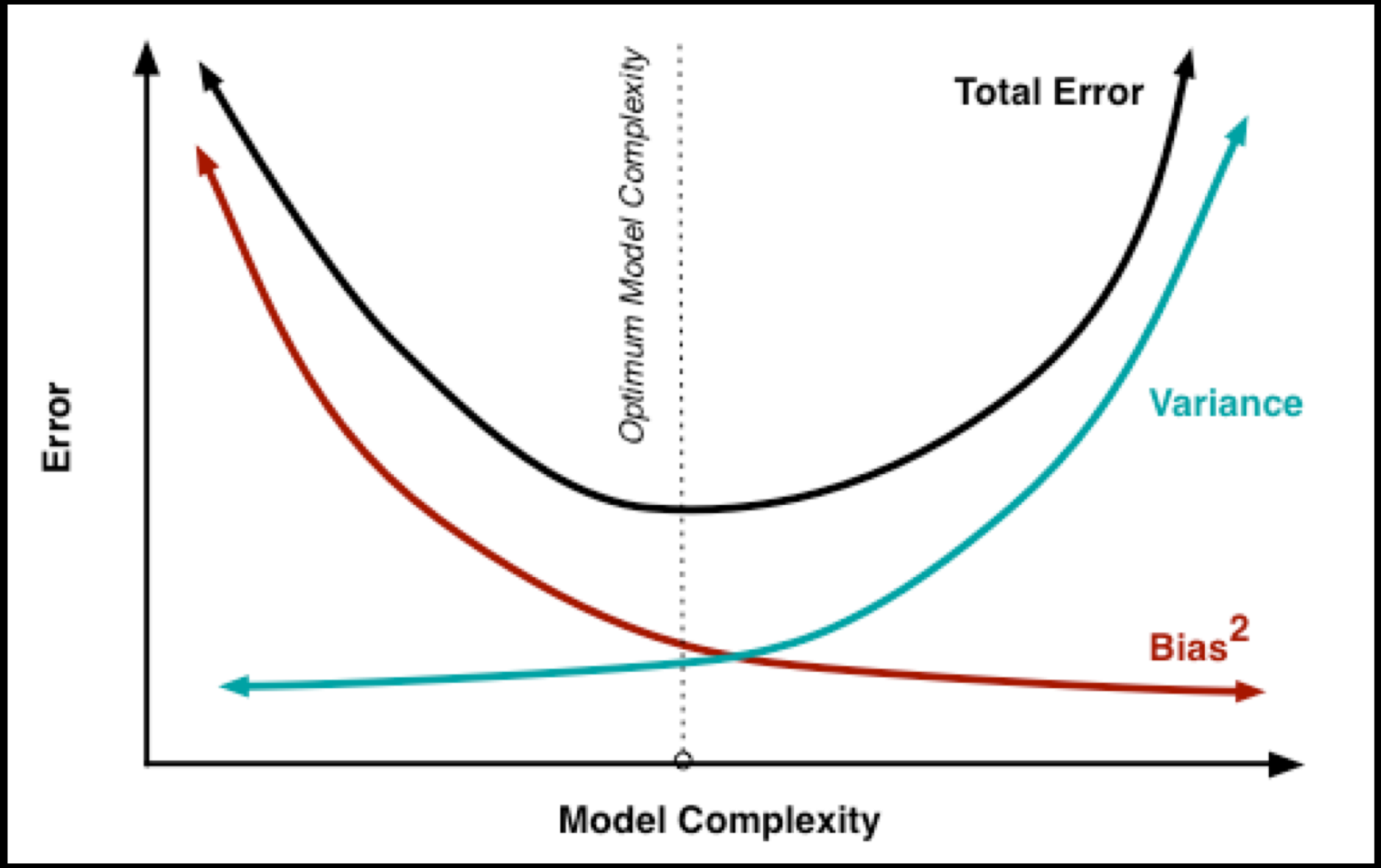
Source: scikit-learn.org

Bias vs. Variance



- To understand overfitting, it's helpful to understand the concepts of bias and variance
- Bias: consistently learned the wrong thing
- Variance: learn random things irrespective of the true signal

Relationship to Overfitting



Underfit vs. Overfit vs. Just Right

Algorithm	Underfit	Overfit	Just Right
kNN	Low k	High k	Reasonable value like 5, 7
Logistic Regression	Linear model	High degree polynomial	Add regularization term
Decision Tree	Small tree	Extremely deep tree, grows until leaf nodes are completely pure	Prune branches where nodes have certain purity

$$\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

Key Takeaways

- Ensure you've set up distinct training, validation and test data
- Don't confuse training set error with test error
- Overfitting is the thing we worry about the most

Unsupervised Learning

Unsupervised Learning

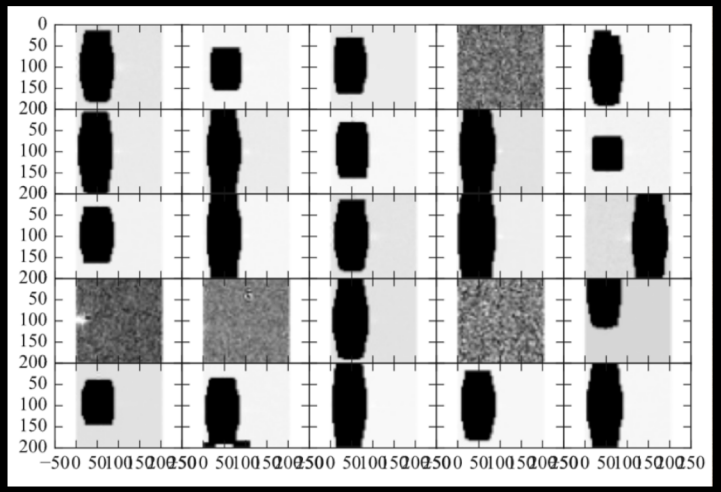
- Learning from data in absence of rewards (reinforcement learning) or labels (supervised learning)
- Major sub-types:
  - Clustering
  - Anomaly Detection
  - Dimensionality Reduction
  - Density Estimation

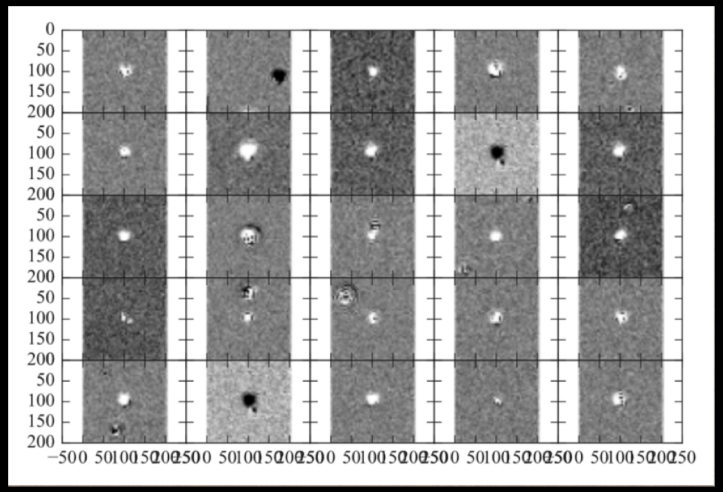
PCA, manifold learning (IsoMap)

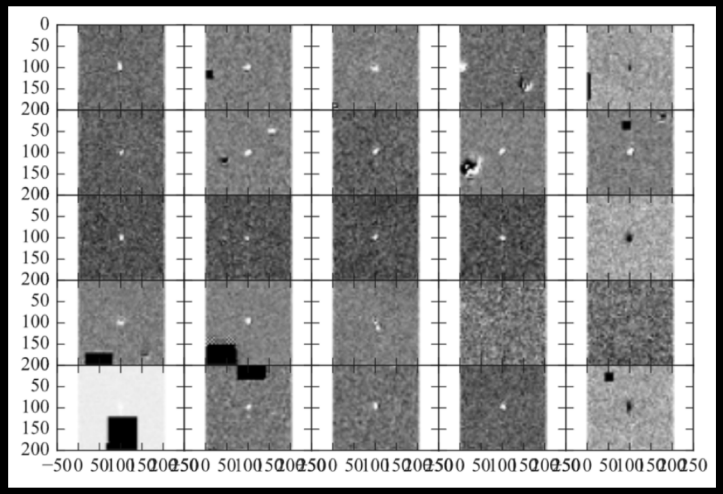
Finding an underlying probability density function

Clustering Example

Understanding Artifacts in ZTF Image Subtractions



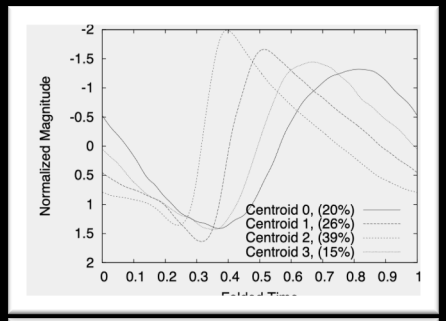


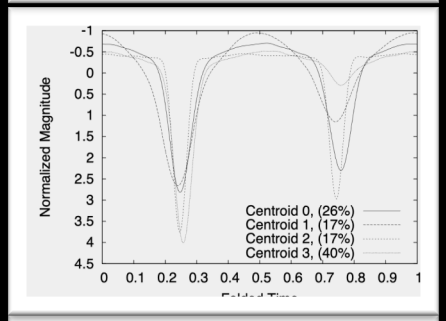


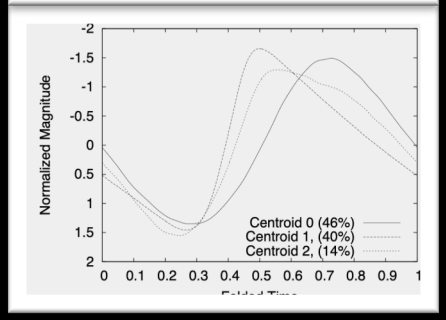
Clustering an early version of training data revealed major classes of artifacts.

Anomaly Detection

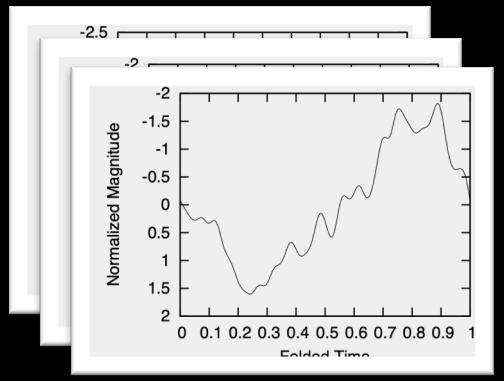
Finding Anomalous Lightcurves in Catalogs of Periodic Variables



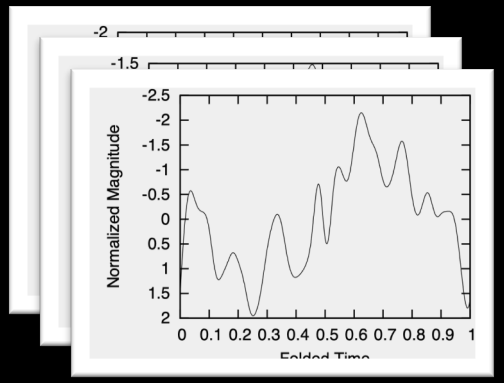


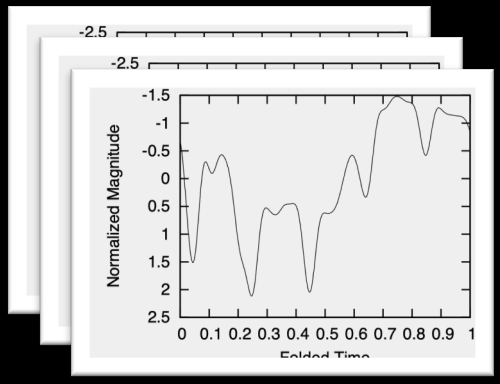


Cluster Centroids (examples of normality)



Anomalies found with respect to these cluster centroids

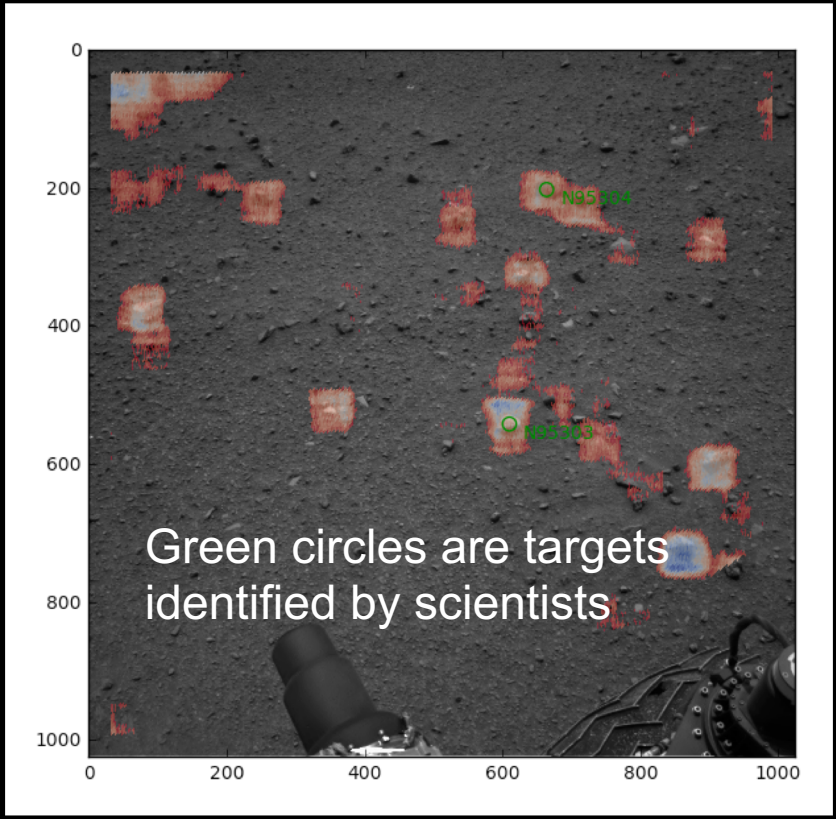


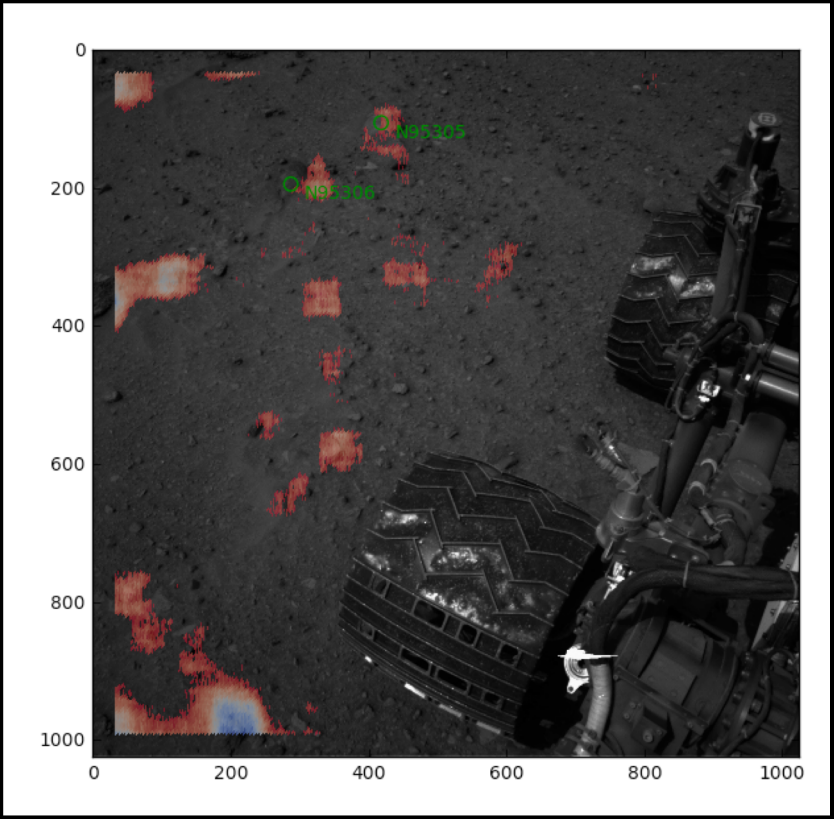


Novelty Detection on MSL Imagery

Navigation Camera Images

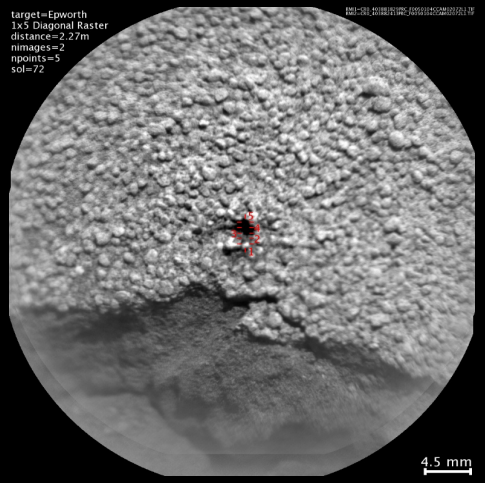
Anomalies identified using Isolation Forest

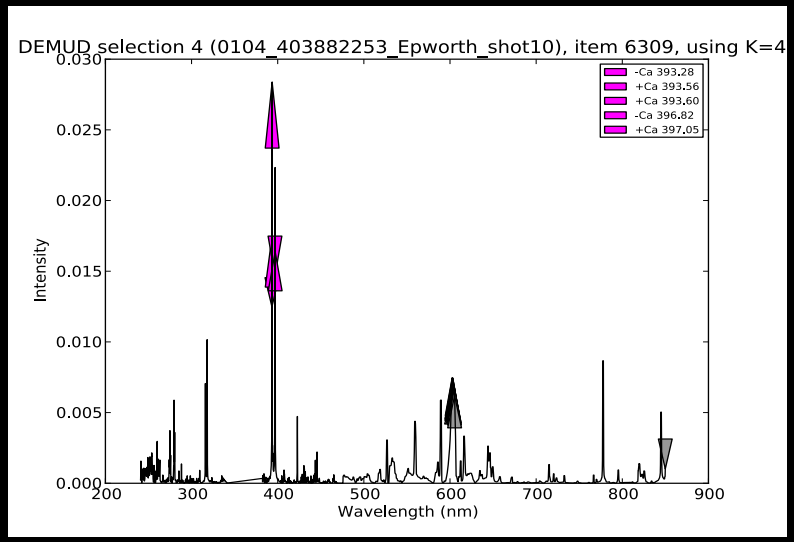




Dimensionality Reduction

Discovery via Eigenbasis Modeling of Uninteresting Data (DEMUD)





A DEMUD result (center) on ChemCam data taken on a soil sample at target Epworth (left). DEMUD found an unexpectedly high occurrence of Ca in this sample (magenta triangles), which turned out to correspond to a scientifically interesting detection of the mineral CaF (grey triangles).

DEMUD uses **singular value decomposition** to model normality in the dataset.

Key Takeaways

- Unsupervised learning constitutes learning without a target concept or reward
- Primary objective is data understanding
- K-means is fast clustering algorithm, but performs poorly in high dimensions and when data is not a mixture of Gaussians with constant variance. Be aware of its biases.
- Anomaly/Outlier detection is very subjective.

Training Set Triage

Something is Not Right

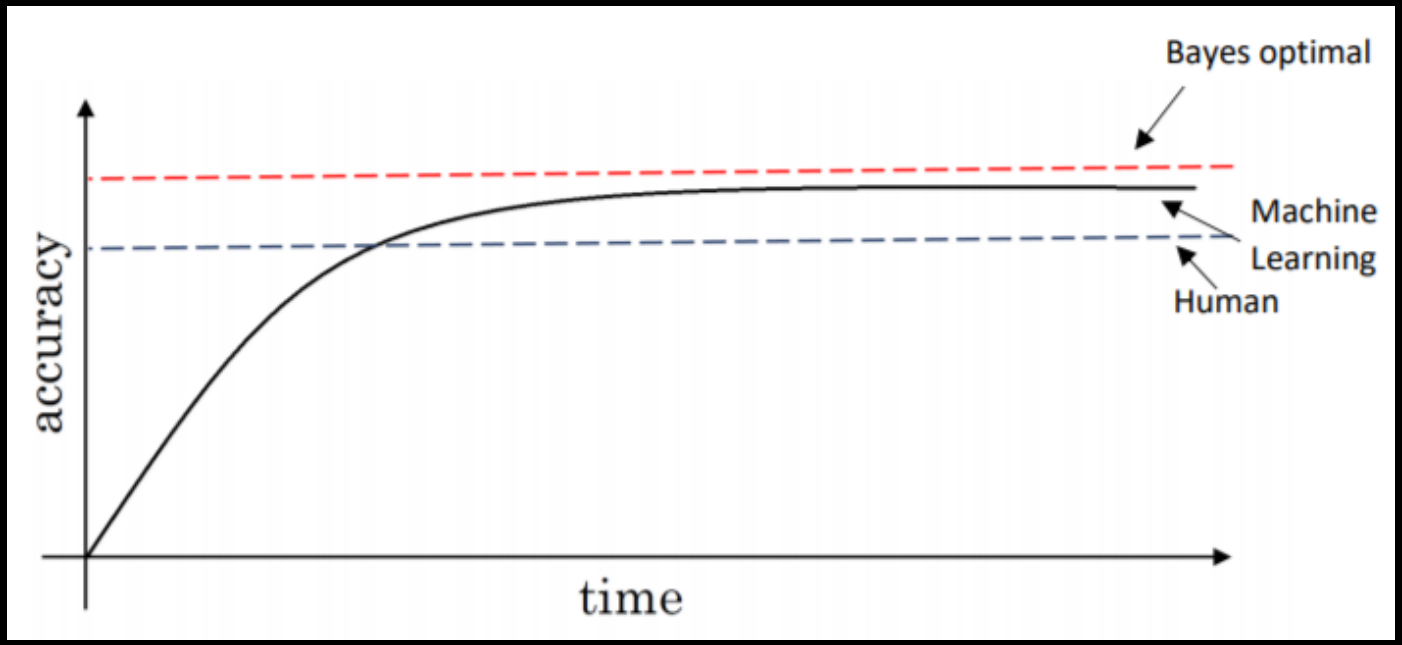
- Performance isn’t acceptable or what you’d hoped for
- Let’s assume that you’ve experimented with different classifiers and you’re using the best performing one.
- How do you debug your machine learning performance?

Outline

- Examine all the ingredients:
  - Metrics
  - Features
  - Examples, including Labels

Metrics

- Bayes Error: best theoretical performance that your classifier can achieve



- How do you know when you've achieved it?

Source: Andrew Ng's Structuring Machine Learning Projects, Coursera

Metrics

- You can't! But perhaps you can use human performance as a proxy for the Bayes error
- Alternatively, if human performance is better than your ML performance, then you have some hope of improving.

Looking at Train / Test Errors

- Consider this data about your classifier

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

- On Scenario A, your training error is much worse than human error. You may have an **avoidable bias**
- On Scenario B, your training error is about the same

Looking at Train / Test Errors

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

- In both cases, the test error is 2% more than training error. This is a sign that you have overfitting.

Looking at Train / Test Errors

	Classification error (%)	
	Scenario A	Scenario B
Humans	1	7.5
Training error	8	8
Development error	10	10

- Bias avoidance strategies: change classifiers
- Variance avoidance strategies: tune hyperparameters, use regularization

Features

- Extreme values
  - are those valid or garbage examples?
- Sentinel values - usually these are stand-ins for NaN
  - Invalid values – could be indicative of a problem
- Check with science teams, pipeline people
  - Just because it can be a feature, doesn’t mean it should
  - Toss out features that are known to be problematic

Examples

- Are there examples left over from engineering or science validation phases of the survey?
- Are your classes balanced? Do you have an extreme minority class?
- **Are your labels contaminated?**

Examples

- Are there examples left over from engineering or science validation phases of the survey?
  - Remove any problematic examples that are “stale”
  - Remove extreme feature values that may be indicative of some type of problem
- Are your classes balanced? Do you have an extreme minority class?
  - Oversample your minority class
  - Undersample your majority class

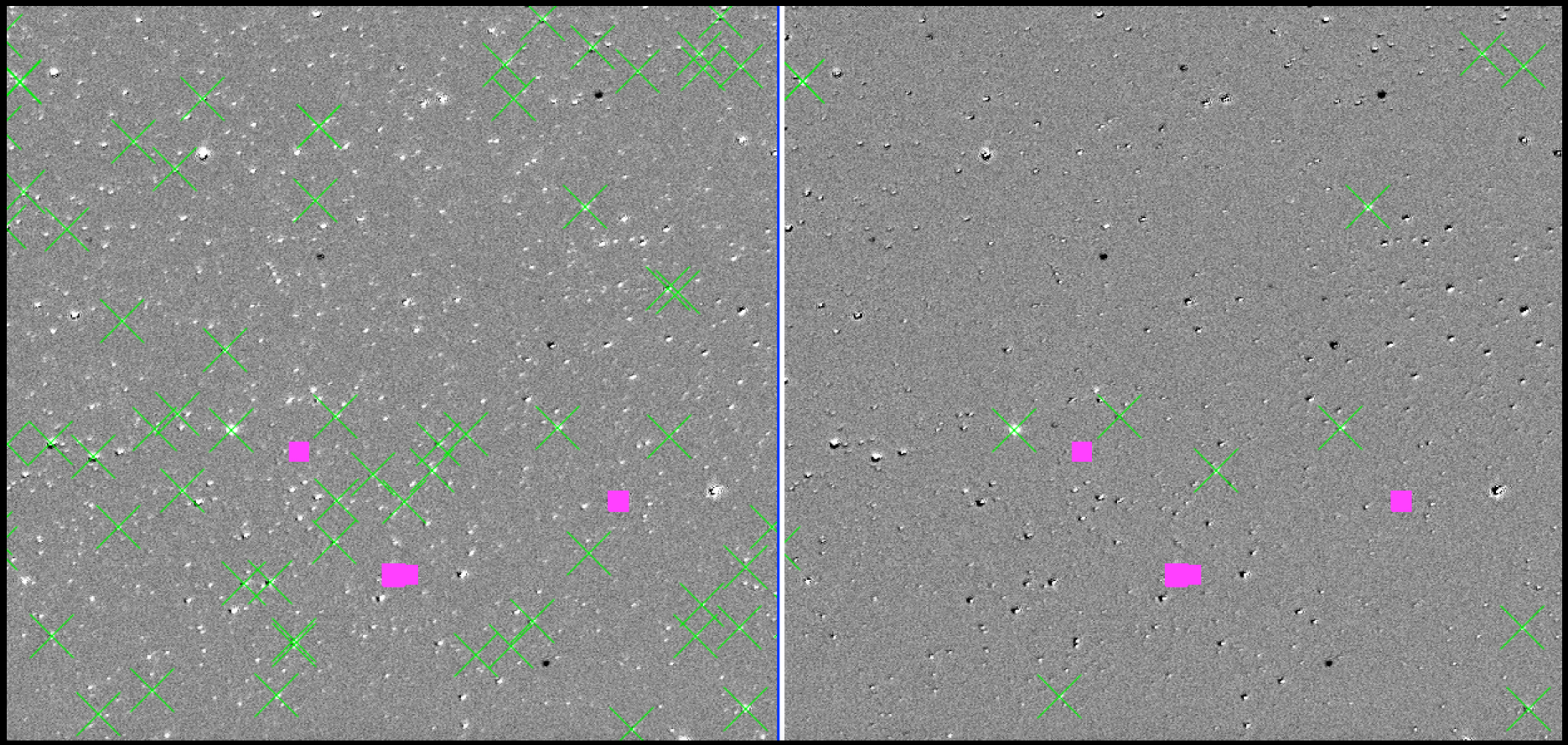
Concept Drift

- When your test distribution starts drifting away from your training distribution
- Why would this happen?

Stuff Happens

- Pipeline upgrades
  - Reference image upgrades
  - Image subtraction changes
- Telescope changes/repairs
- Survey priorities change (e.g., asteroid and GP surveys)
- No one tells the ML team

Image Subtraction Upgrade

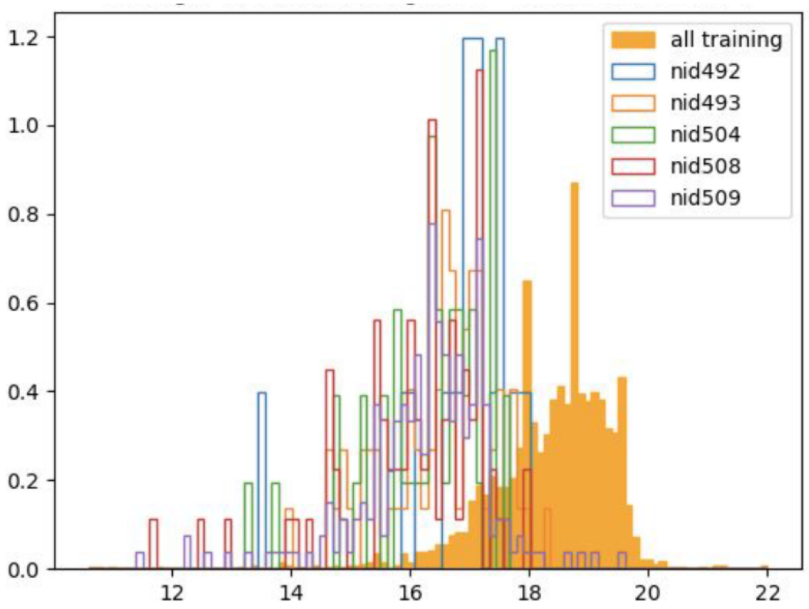


Before

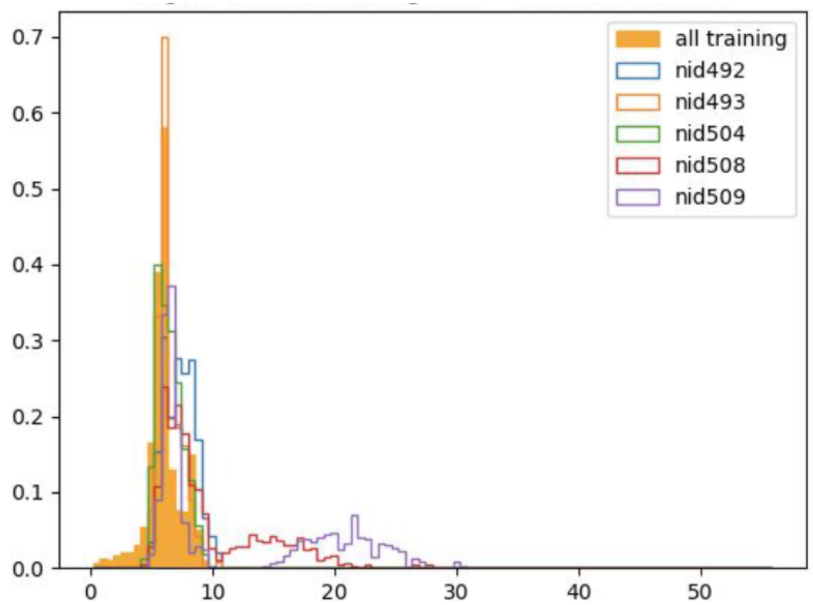
After

Checking Sample Bias

Ssmagnr
Magnitude of nearest solar system object



Scigain
Electronic gain in science-image (after gain-matching)

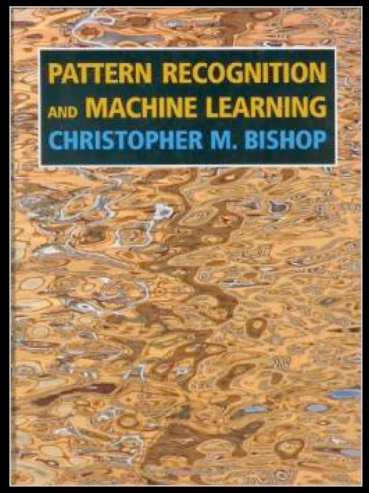


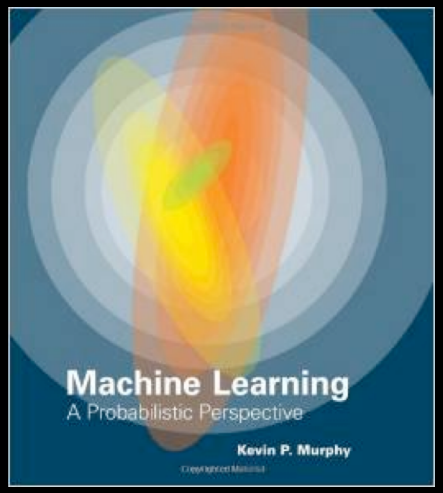
Key Takeaways

- Hard to know what optimal performance is
- Looking at training error versus test error can be useful in determining whether bias or variance may be the issue
- Be mindful of data preprocessing, concept drift, label contamination, etc. These may be where you spend most of your time.

Machine Learning Resources

- Textbooks

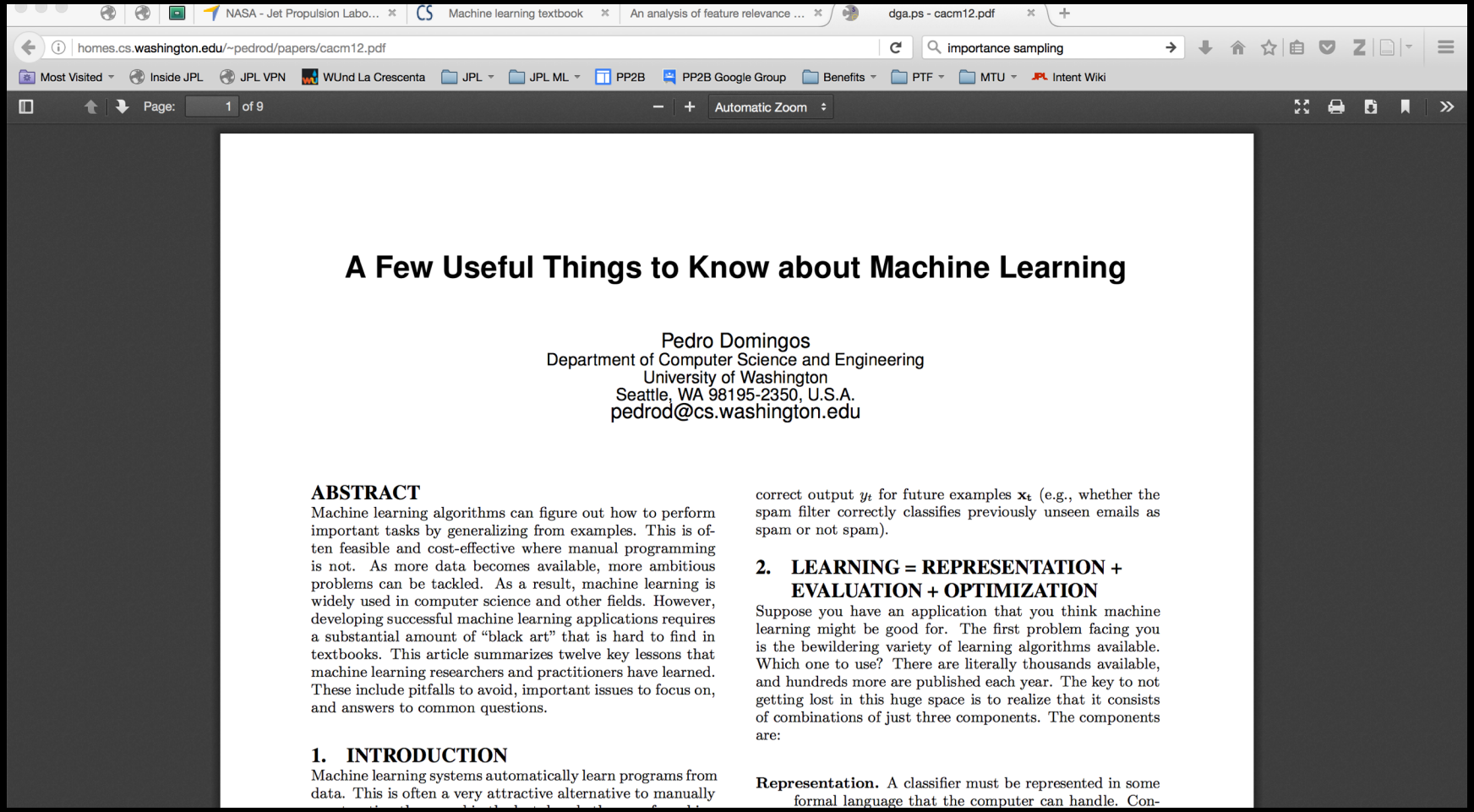




- Christopher Olah's blog: https://colah.github.io/

- Massive Open Online Courses (MOOCs)
  - Coursera: Intro to ML (Prof. Andrew Ng)
  - Coursera: Structuring ML Projects (Prof. Andrew Ng)

“Black Art” of Machine Learning



A Few Useful Things to Know about Machine Learning

Pedro Domingos
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
pedrod@cs.washington.edu

ABSTRACT

Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. As more data becomes available, more ambitious problems can be tackled. As a result, machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of “black art” that is hard to find in textbooks. This article summarizes twelve key lessons that machine learning researchers and practitioners have learned. These include pitfalls to avoid, important issues to focus on, and answers to common questions.

1. INTRODUCTION

Machine learning systems automatically learn programs from data. This is often a very attractive alternative to manually

correct output y_t for future examples $\mathbf{x_t}$ (e.g., whether the spam filter correctly classifies previously unseen emails as spam or not spam).

2. LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Suppose you have an application that you think machine learning might be good for. The first problem facing you is the bewildering variety of learning algorithms available. Which one to use? There are literally thousands available, and hundreds more are published each year. The key to not getting lost in this huge space is to realize that it consists of combinations of just three components. The components are:

Representation. A classifier must be represented in some formal language that the computer can handle. Con-

Expanded Version of These Slides

- Includes more basic tutorials on ML and also Deep Learning
- Data and Jupyter notebooks to play with
- https://github.com/LSSTC-DSFP/LSSTC-DSFP-Sessions/tree/master/Session7

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov