Supervised Machine Learning

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

LSSTC Data Science Fellowship Program
Thursday, November 8, 2018
Northwestern University

Jet Propulsion Laboratory
California Institute of Technology

Outline

- Ingredients
- Ingredients In-depth:
  - Data Sampling
  - Learning Algorithms
  - Evaluation
- Overfitting and Other Key Concepts
- Summary

Ingredients

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

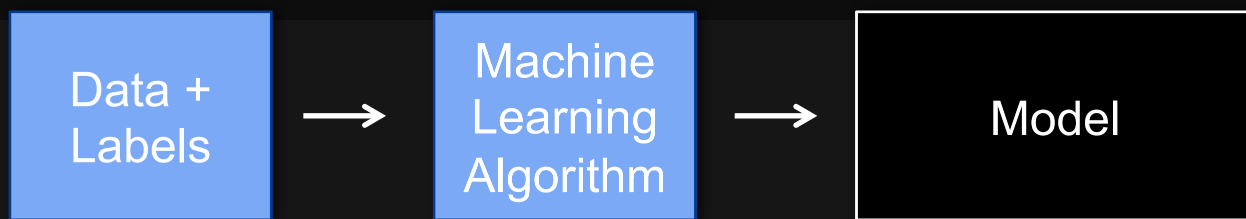
Features

Class label

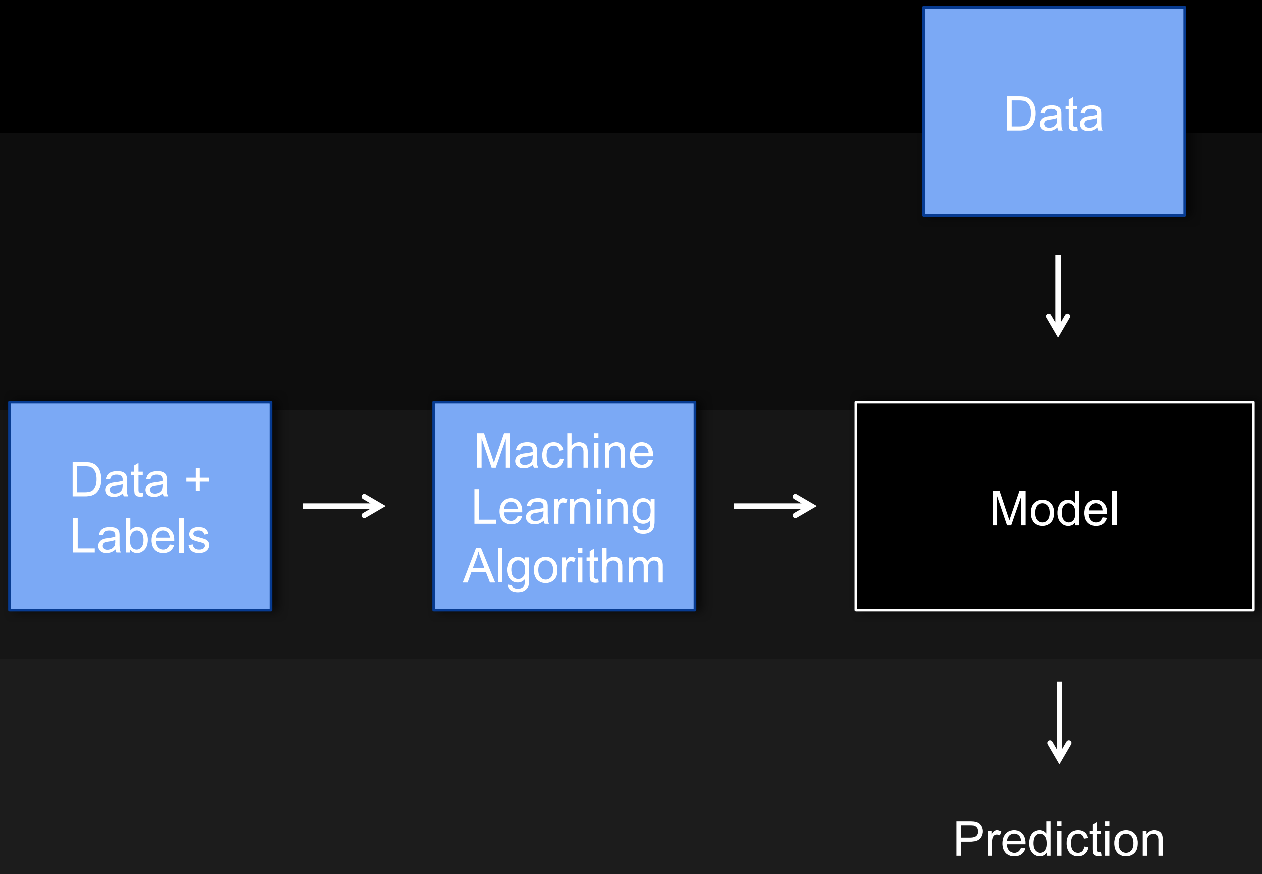
Examples

	# 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						

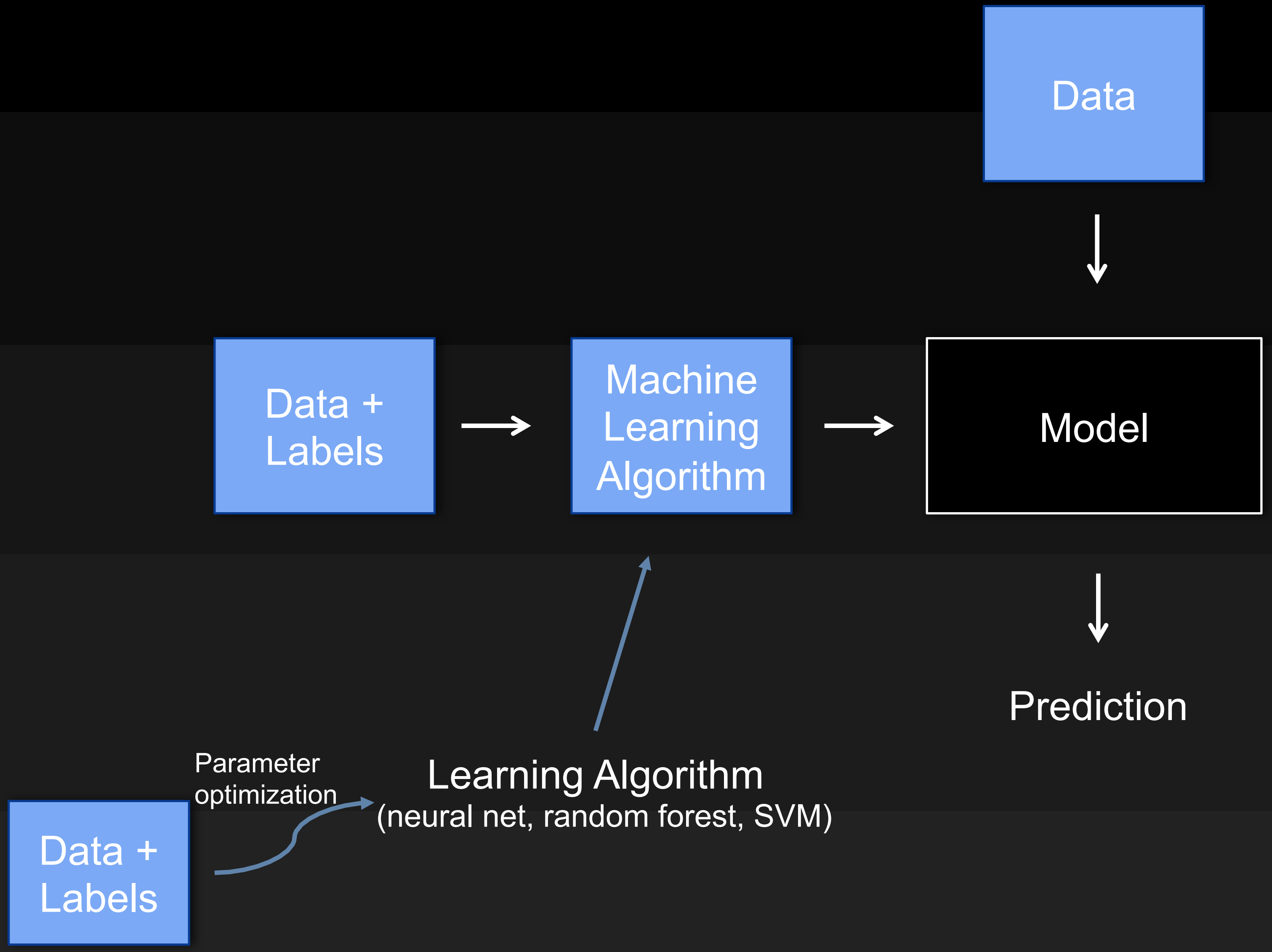
Training a Classifier



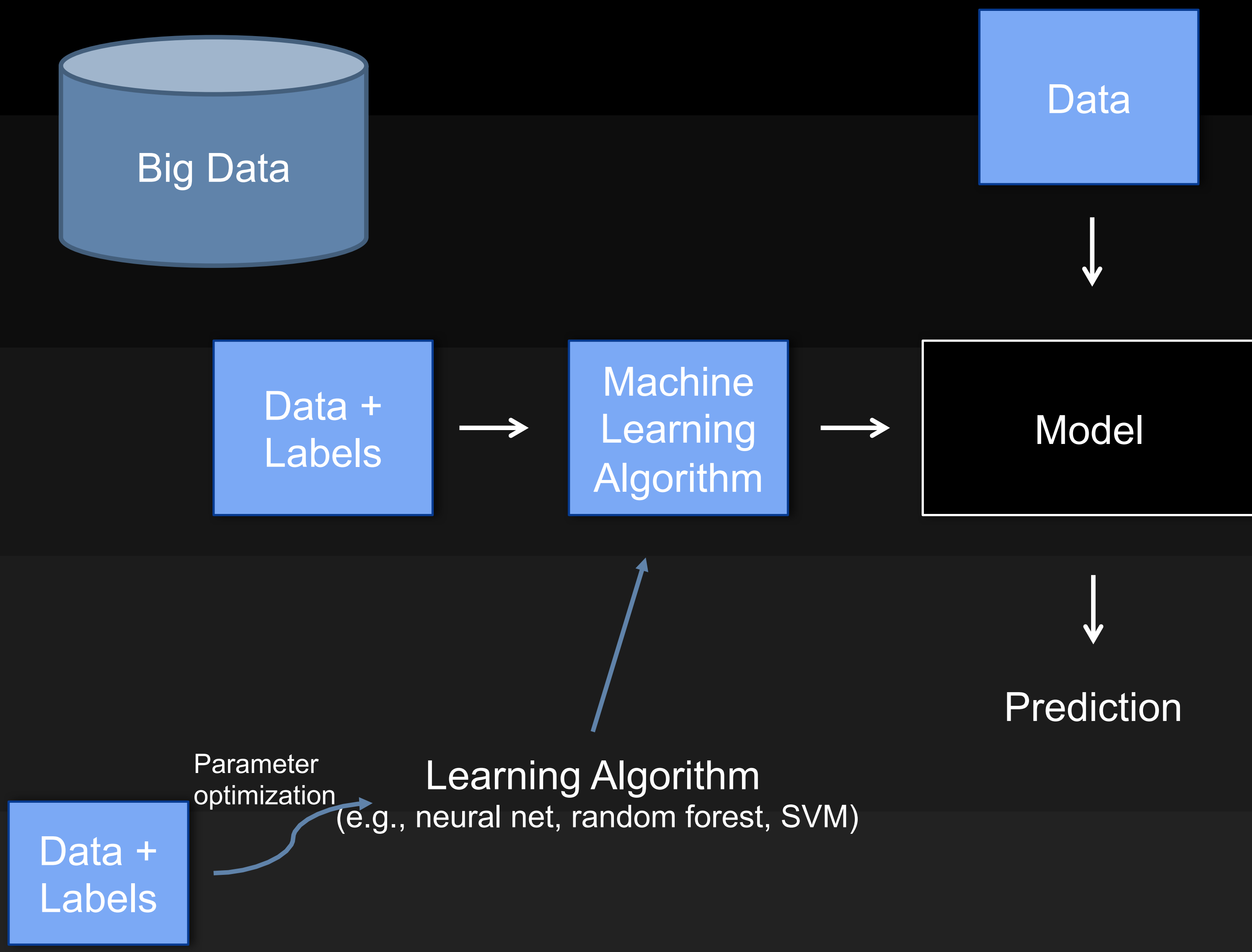
Training a Classifier



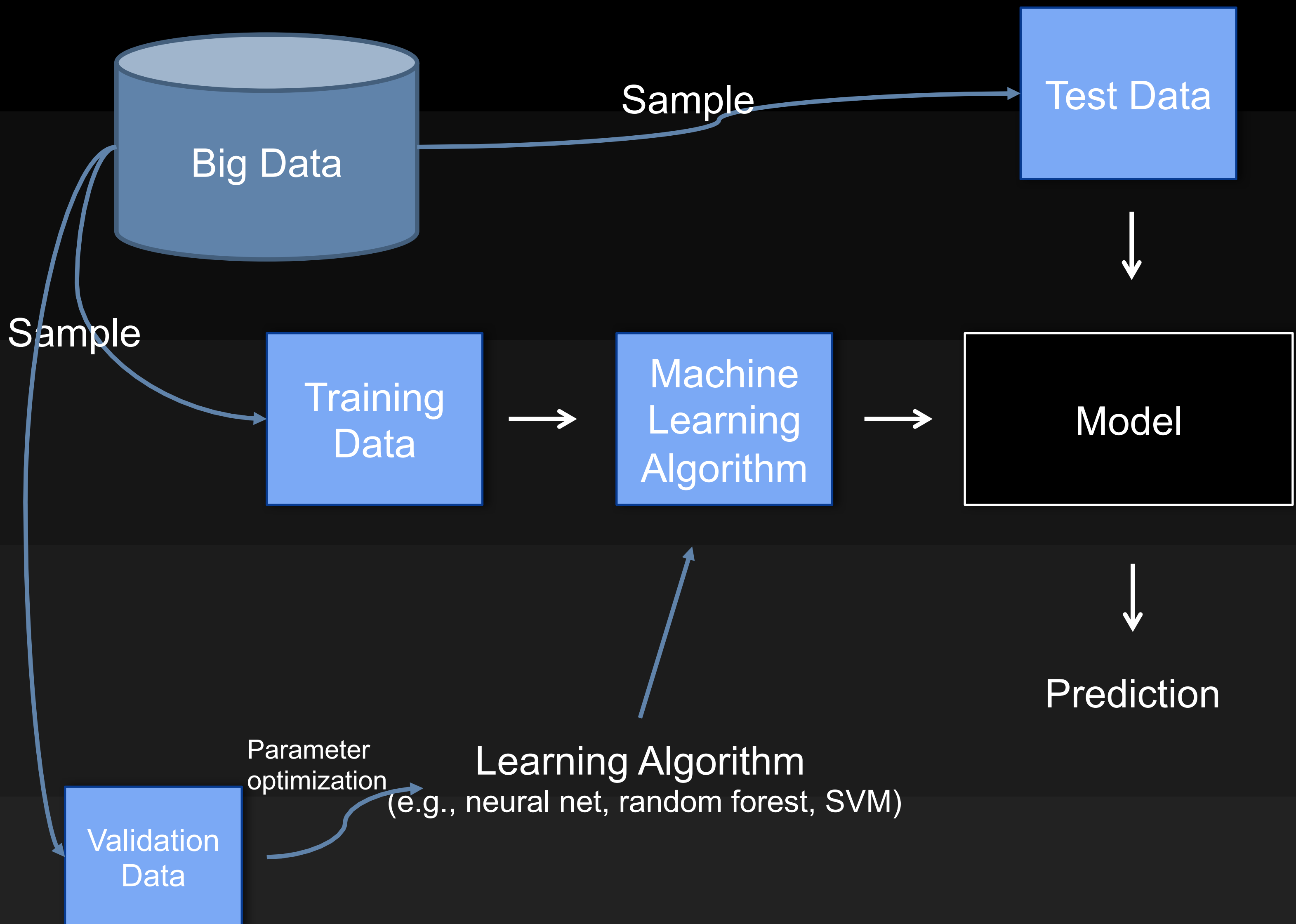
What are the Ingredients?



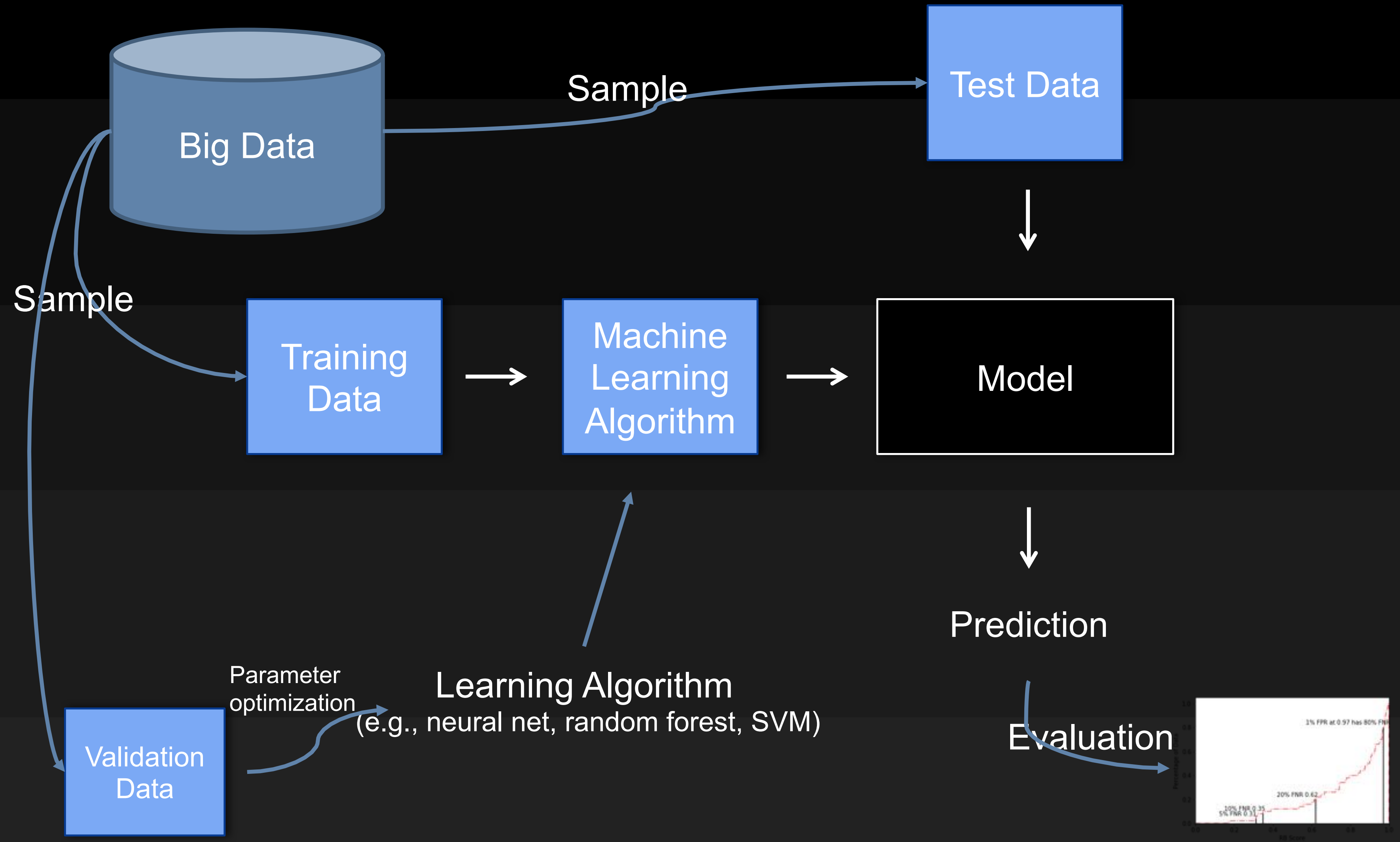
What are the Ingredients?



What are the Ingredients?



What are the Ingredients?



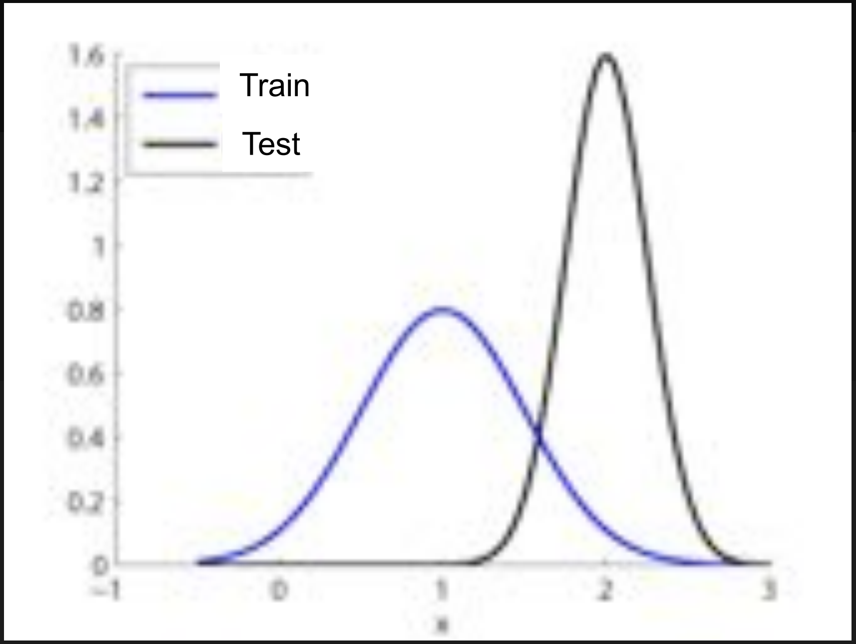
Ingredients Summarized

- Sampling Data into Training, Validation, and Test Sets
- Feature Representation - YESTERDAY
- 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

Labeled Data

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

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.

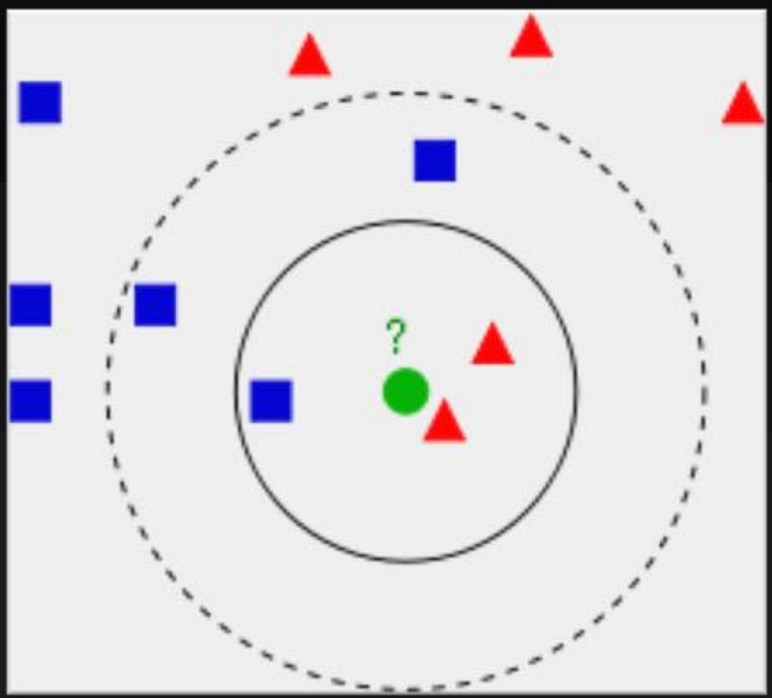
Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic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

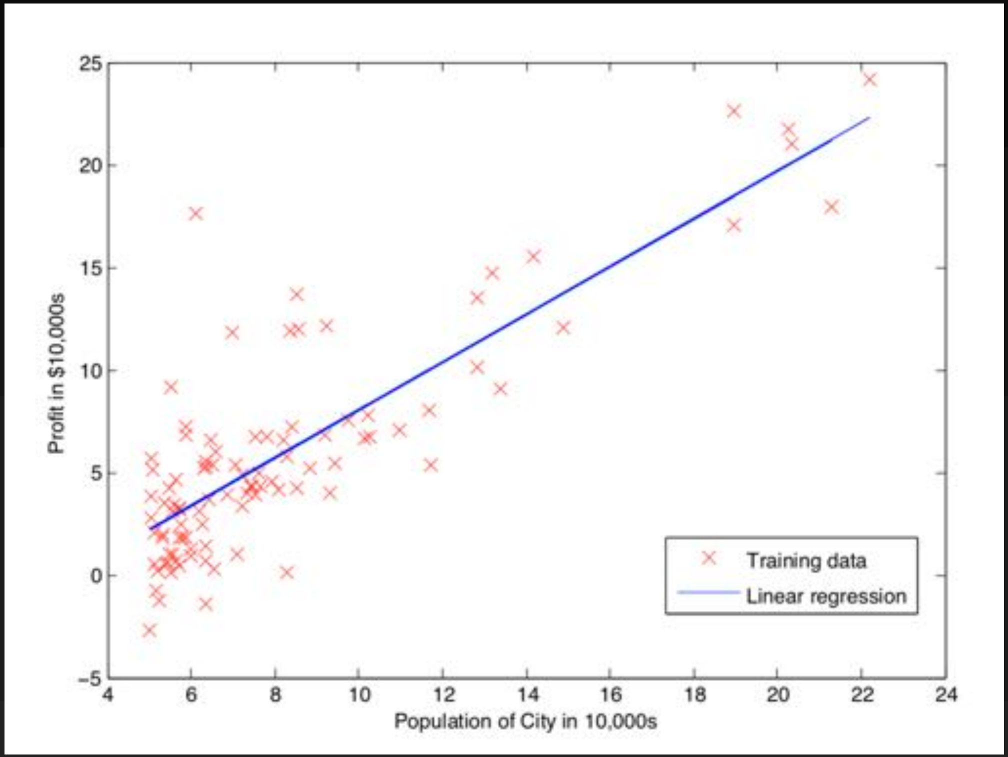
Algorithm	Representation	Evaluation	Optimization
kNN	Example	Squared Distance	Greedy Search
Logistic Regression			
Decision Tree			

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

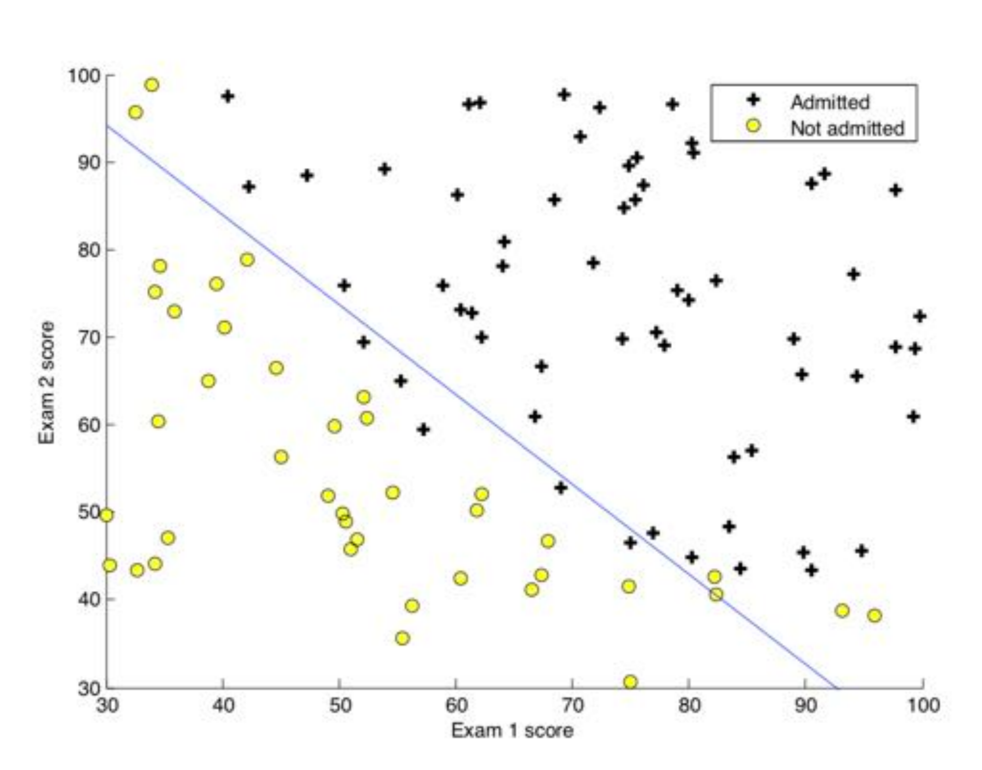
Logistic Regression

- 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.

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

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

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



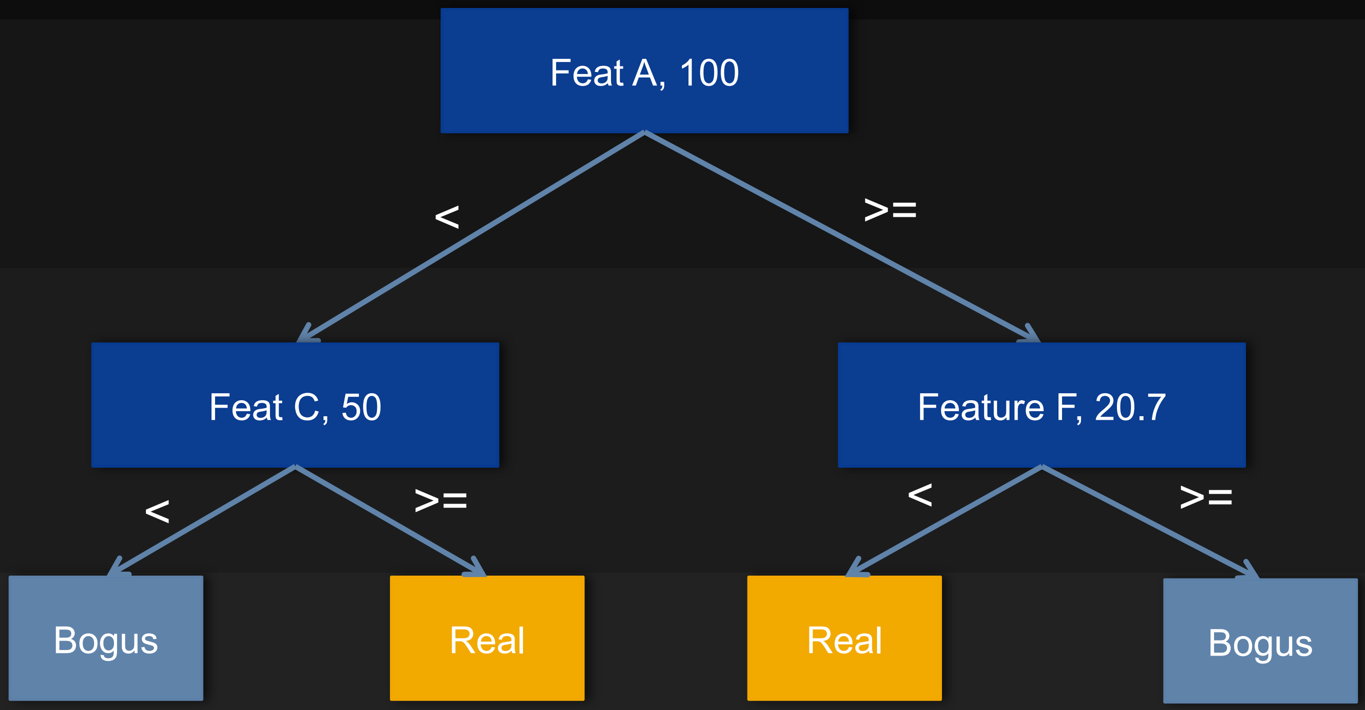
Source: Andrew Ng, Introduction to Machine Learning, Coursera

Logistic Regression

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

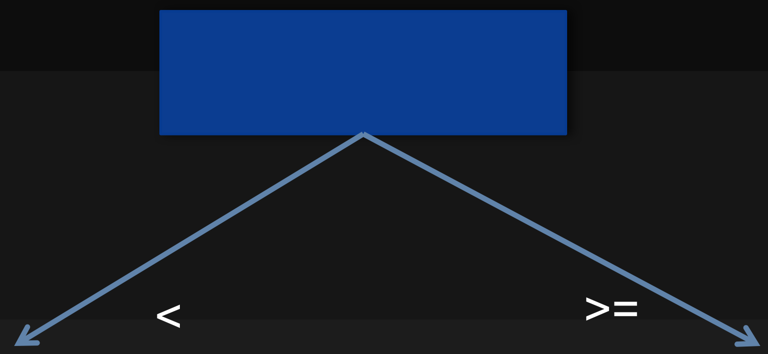
Decision Tree

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



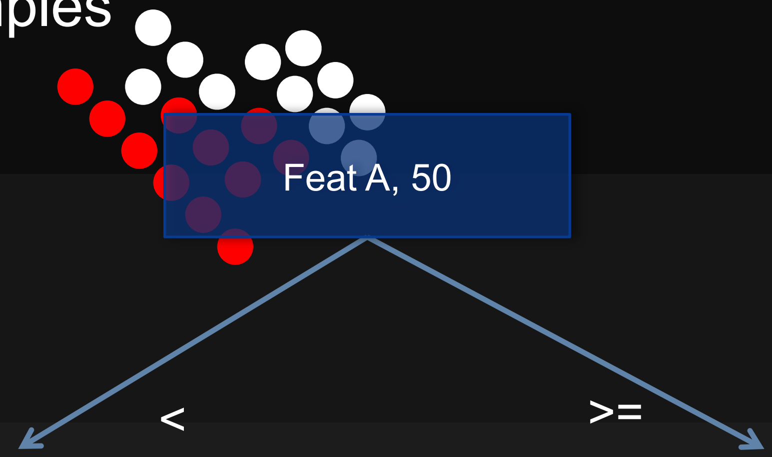
Training a Decision Tree

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



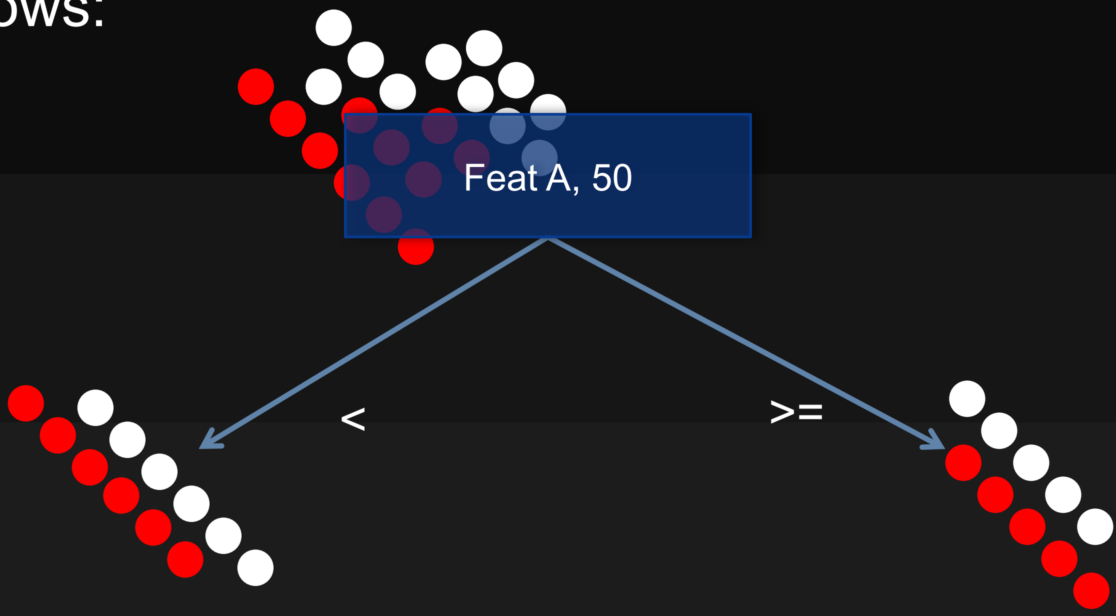
Training a Decision Tree

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



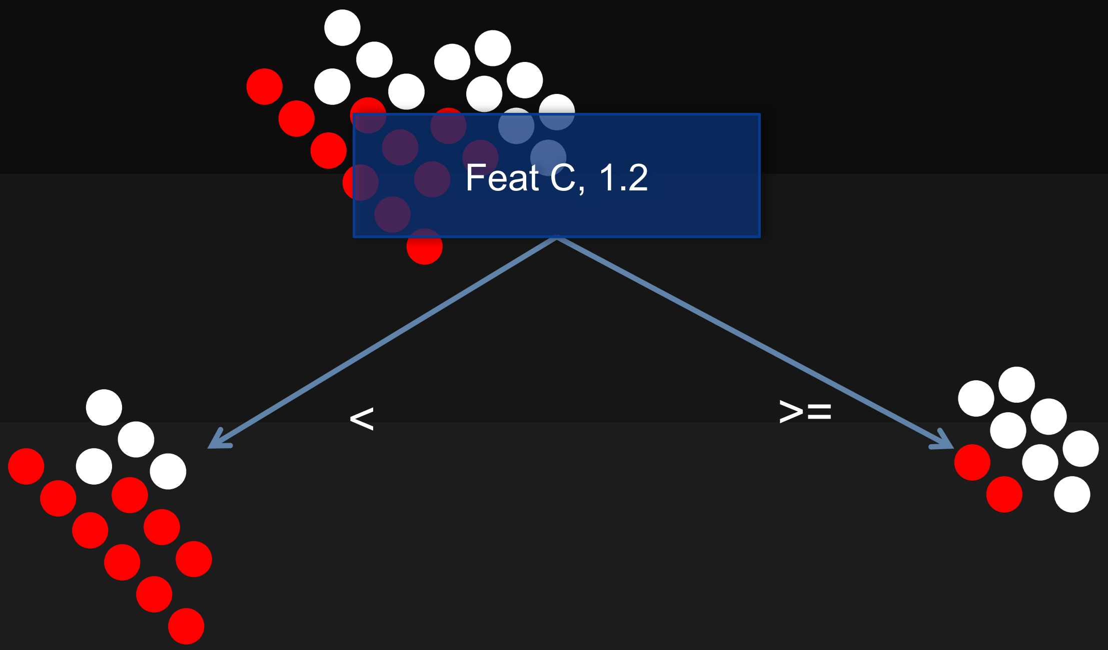
Training a Decision Tree

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



Training a Decision Tree

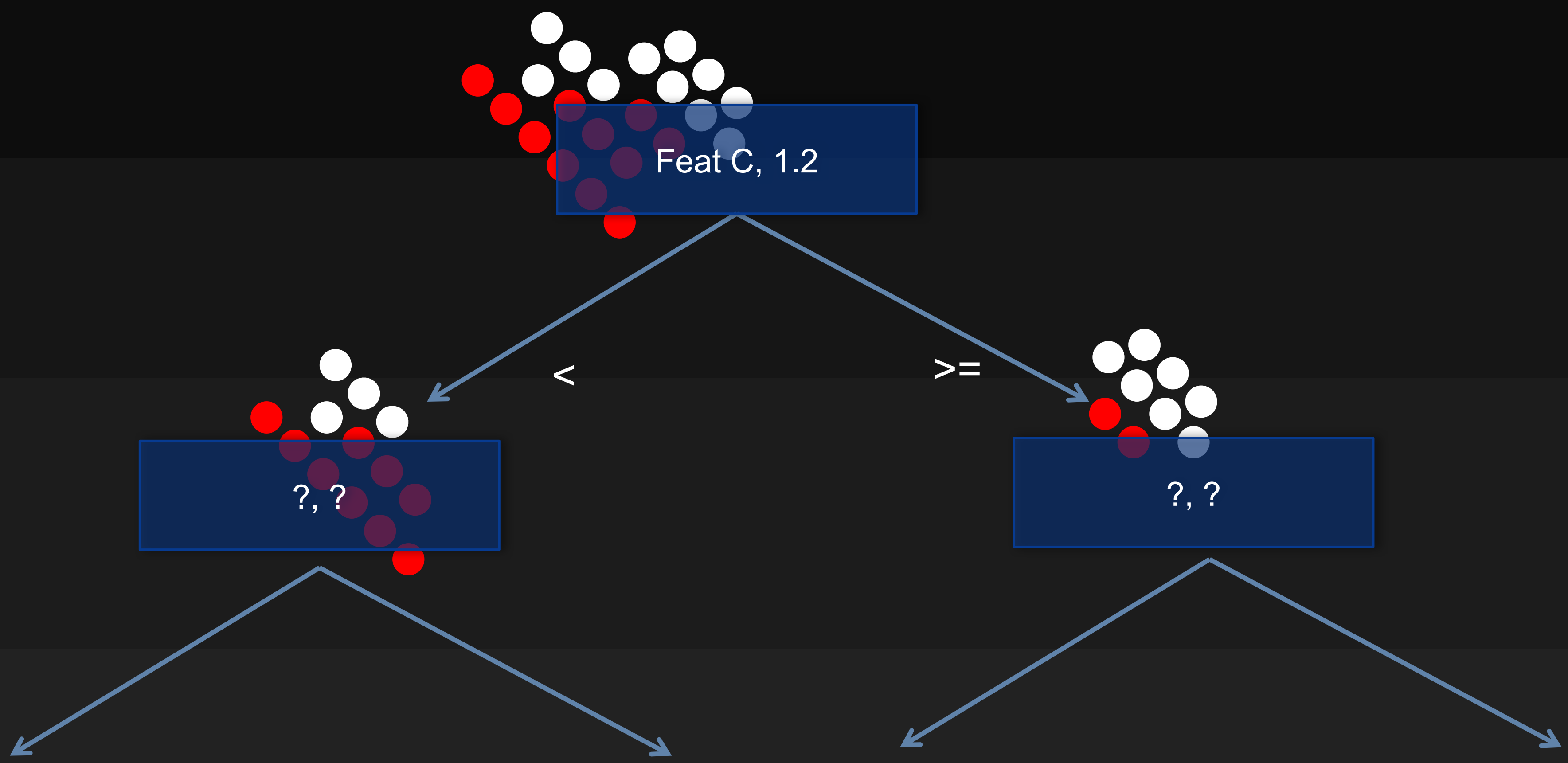
- This feature, feature value pair partitions my training samples as follows:



- Which split is preferable?

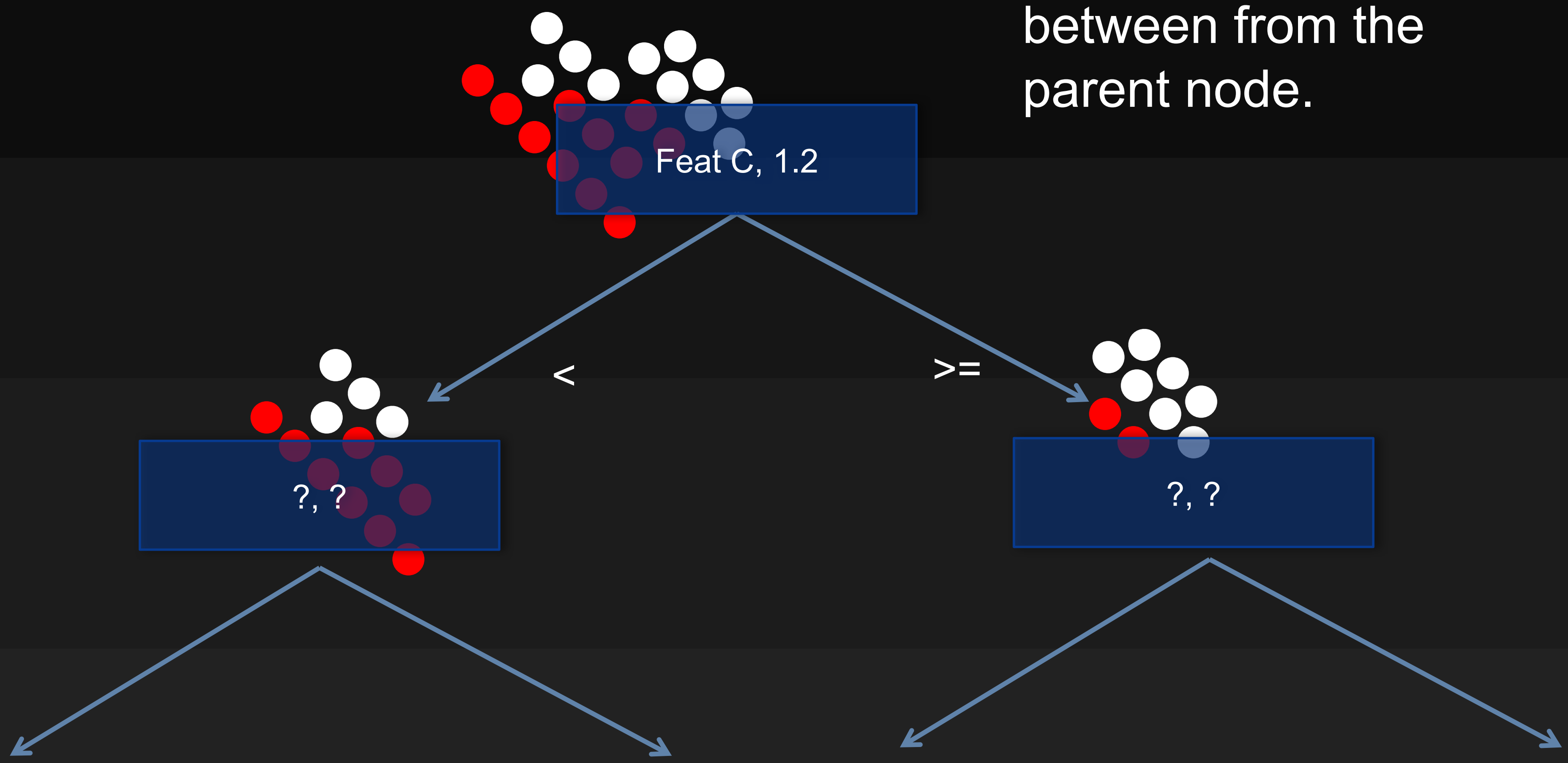
Training a Decision Tree

- 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

Score = 0.54
13 of 24 trees voted Real

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)

Predicted

Actual

	Positive (1)	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)

Predicted

Actual

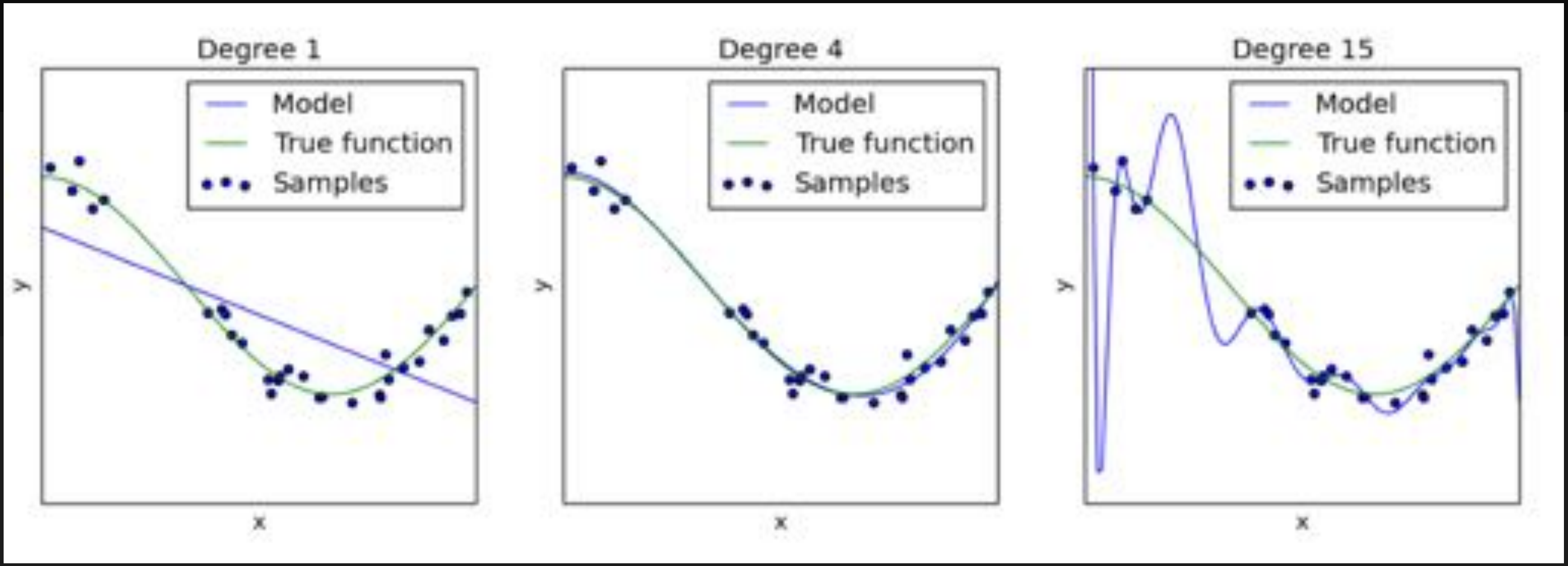
	Positive (1)	Negative (0)
Positive (1)	**True Positive (TP)**	**False Negative (FN)**
Negative (0)	False Positive (FP)	True Negative (TN)

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 an high test error is said to be **overfit**.

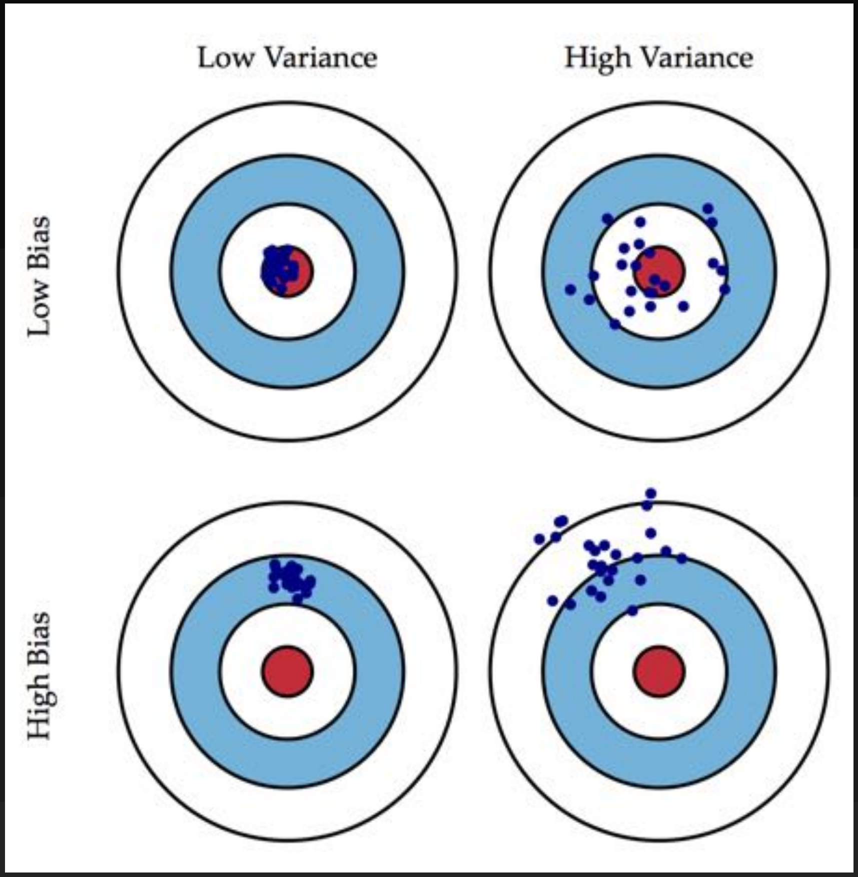
Underfitting vs. Overfitting



Underfitting **Overfitting**

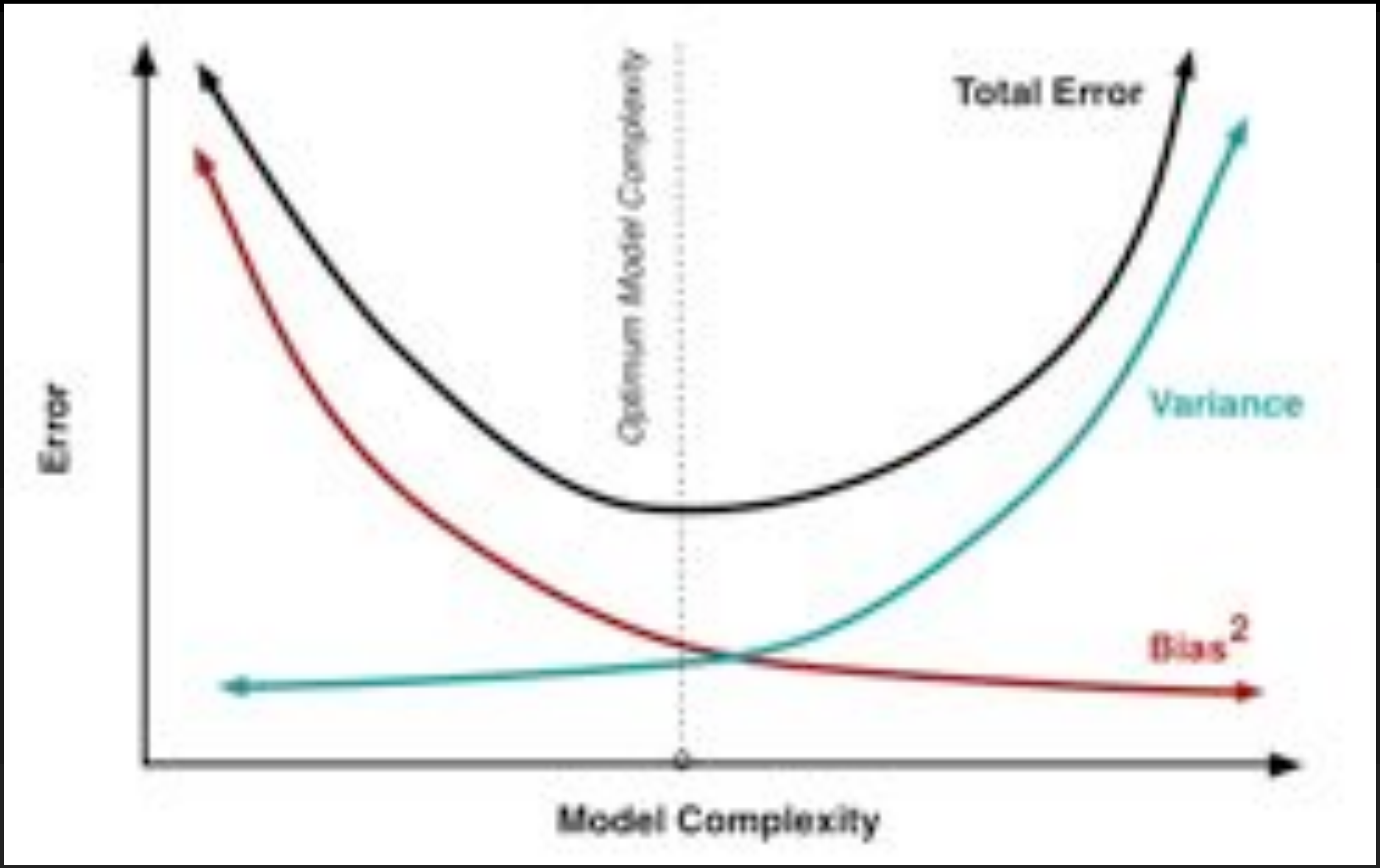
Source: scikit-learn.org

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

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

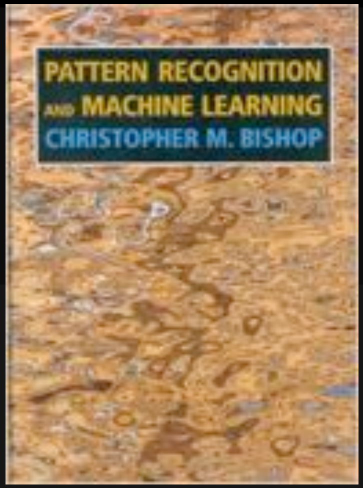
Summary

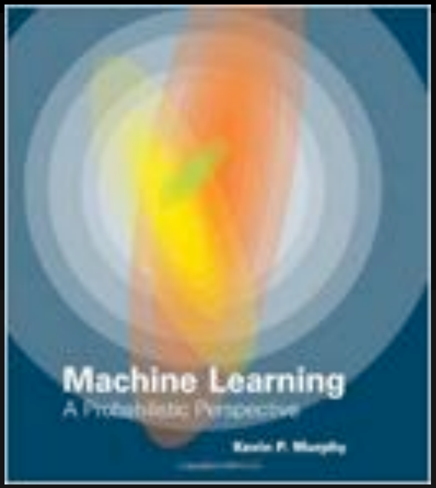
Key Takeaways

- 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

Machine Learning Resources

- Textbooks





- scikit-learn.org

- 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 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-

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov