

# MAS DSE 230

## Scalable Analytics

## Big Data Analytics

Mai H. Nguyen

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- Modeling
- Spark MLlib
- Assignments

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- Modeling
- Spark MLlib
- Assignments

# WHAT IS MACHINE LEARNING?

“... a subfield of computer science that ... explores the study and construction of algorithms that can learn from and make predictions on data.” ([wikipedia.org](https://en.wikipedia.org))

“... a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed.” ([whatis.techtarget.com](https://whatis.techtarget.com))

“... a method of data analysis that automates analytical model building and ... allows computers to find hidden insights to produce ... predictions that can guide better decisions and smart actions...” ([www.sas.com](https://www.sas.com))

# WHAT IS MACHINE LEARNING?

learning from data

no explicit programming

discover hidden patterns

data-driven decisions

# WHAT IS MACHINE LEARNING?

learning from data

no explicit programming

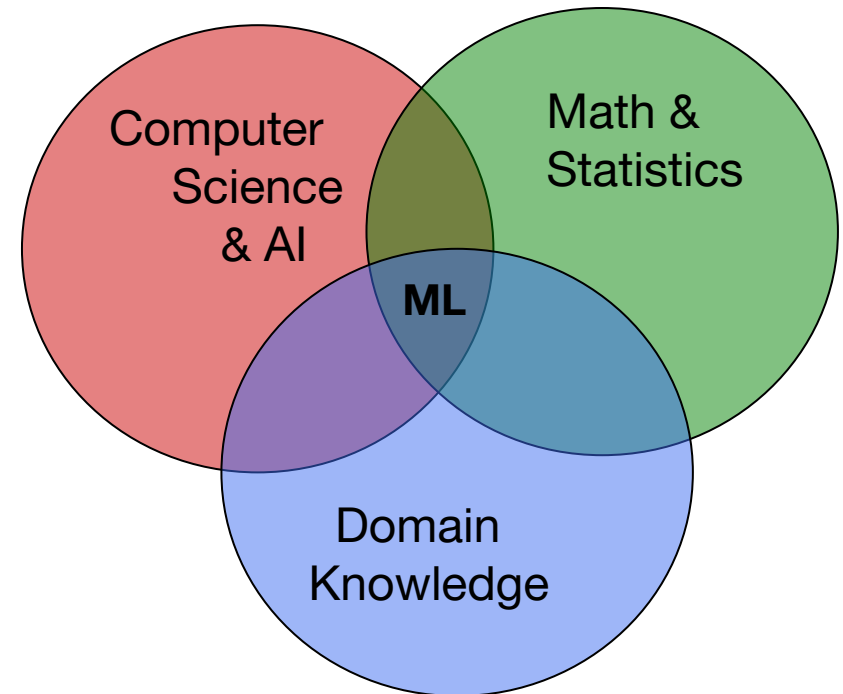


# WHAT IS MACHINE LEARNING?

- Working Definition
  - The field of machine learning focuses on the study and construction of computer systems that can learn from data without being explicitly programmed. Machine learning algorithms and techniques are used to build models to discover hidden patterns and trends in the data, allowing for data-driven decisions to be made.

# MACHINE LEARNING IS INTERDISCIPLINARY

- ML combines concepts & methods from many disciplines:
  - Mathematics, statistics, computer science, optimization, etc.
- ML has been used in various applications:
  - Science, engineering, business, medical, law enforcement, etc.





# MACHINE LEARNING APPLICATIONS

## Best Sellers based on your browsing history



Apple AirPods with Charging Case (Wired)  
★★★★★ 153,701  
\$129.00



Apple AirPods Pro  
★★★★★ 54,773  
\$219.00



Apple EarPods with Lightning Connector - White  
★★★★★ 38,539  
\$19.98



Apple AirPods with Wireless Charging Case  
★★★★★ 24,208  
\$159.99



TOZO T10 Bluetooth 5.0 Wireless Earbuds with Wireless Charging Case IPX8 Waterproof TWS...  
★★★★★ 107,951  
\$29.98

## Inspired by your browsing history



AirPods Case Cover with Keychain, Full Protective Silicone AirPods Accessories Skin Cover...  
★★★★★ 18,919  
\$5.99



Apple Watch Series 3 (GPS, 38mm) - Space Gray Aluminum Case with Black Sport Band  
★★★★★ 49,269  
\$169.00



AirPods Case, GMYLE Silicone Protective Shockproof Case Cover Skins with Keychain...  
★★★★★ 15,592  
\$5.98



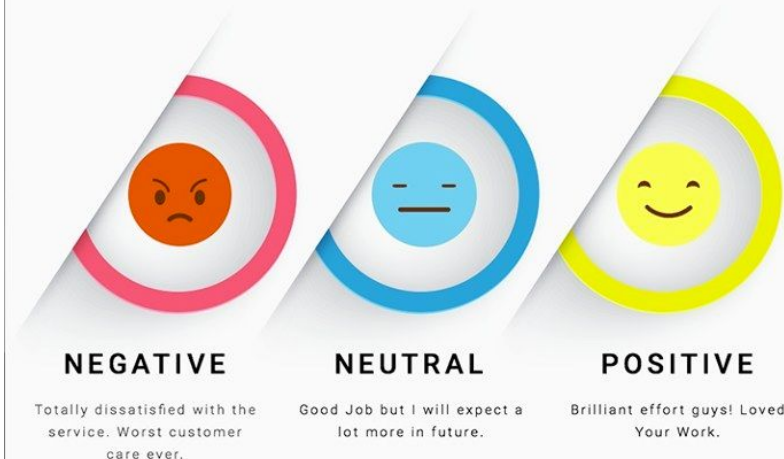
Apple 5W USB Power Adapter  
★★★★★ 3,627  
\$16.99



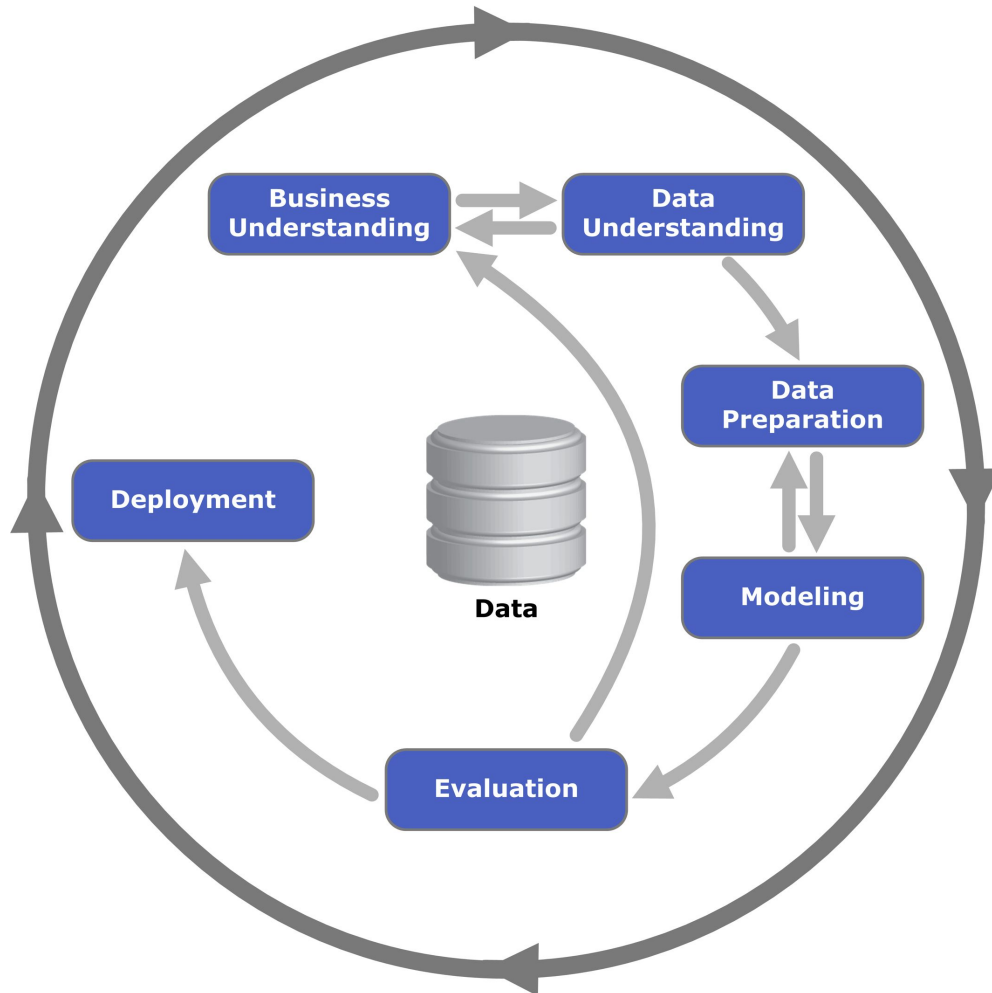
AmazonBasics Premium AirPods Case - Compatible with Apple AirPods 1 & 2, Pink  
★★★★★ 78  
\$6.77



## SENTIMENT ANALYSIS



# MACHINE LEARNING PROCESS



Cross Industry  
Standard Process for  
Data Mining

[https://en.wikipedia.org/wiki/Cross\\_Industry\\_Standard\\_Process\\_for\\_Data\\_Mining](https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining)

# PHASE 1: BUSINESS UNDERSTANDING

- Define problem or opportunity
  - What is the problem of interest? Why is it interesting?
- Assess situation
  - Resources
  - Requirements, assumptions, and constraints
  - Risks and contingencies; costs and benefits
- Formulate goals and objectives
  - Goals and objectives
  - Success criteria
- Create project plan
  - Steps to achieve goals



<https://www.lionessesofafrica.com/blog/2017/3/1/understanding-the-difference-between-a-business-strategy-vs-a-marketing-strategy>

# PHASE 2: DATA UNDERSTANDING

- Data Acquisition
  - Collect available data related to problem.
  - Consider all sources: flat files, databases, sensors, websites, etc.
  - Integrate data from multiple sources
- Exploratory Data Analysis
  - Preliminary exploration of data
  - To become familiar with data
  - Use summary statistics, visualization



<http://www.greenbookblog.org/2013/08/04/50-new-tools-democratizing-data-analysis-visualization/>

# PHASE 3: DATA PREPARATION

- Goal:

- Prepare data to make it suitable for modeling
- Also referred to as 'data preprocessing', 'data munging', 'data wrangling'

- Activities:

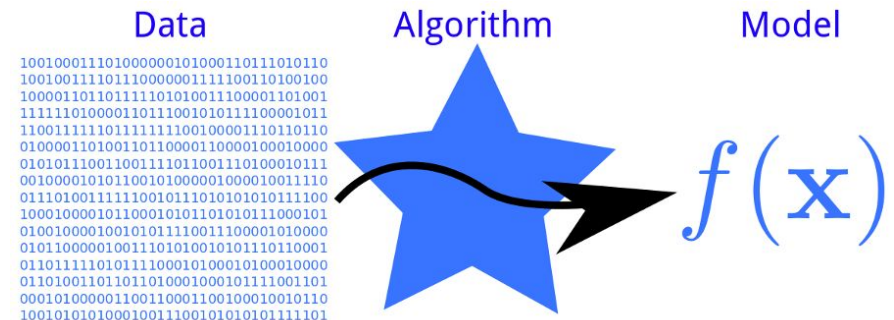
- Identify and address quality issues
- Select features to use
- Create data for modeling



<http://www.datasciencecentral.com/profiles/blogs/5-data-cleansing-tools>

# PHASE 4: MODELING

- Determine type of problem
  - Classification
  - Regression
  - Cluster analysis
- Build model(s)
  - Select modeling technique(s) to use
  - Construct model(s)
  - Train model(s)



<http://phdp.github.io/posts/2013-07-05-dtl.html>

# PHASE 5: EVALUATION

- Assess model performance
  - Determine metrics & methods to assess model results
    - Accuracy measures, confusion matrix, etc.
  - Evaluate model results w.r.t. success criteria.
    - Does model's performance meet success criteria?
    - Have all requirements been met?
- Make Go/No-Go decision
  - Go: Deploy model
  - No-Go: Determine next steps

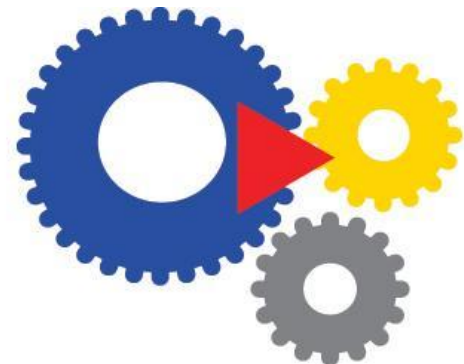


<http://www.impactptac.com/?id=10>



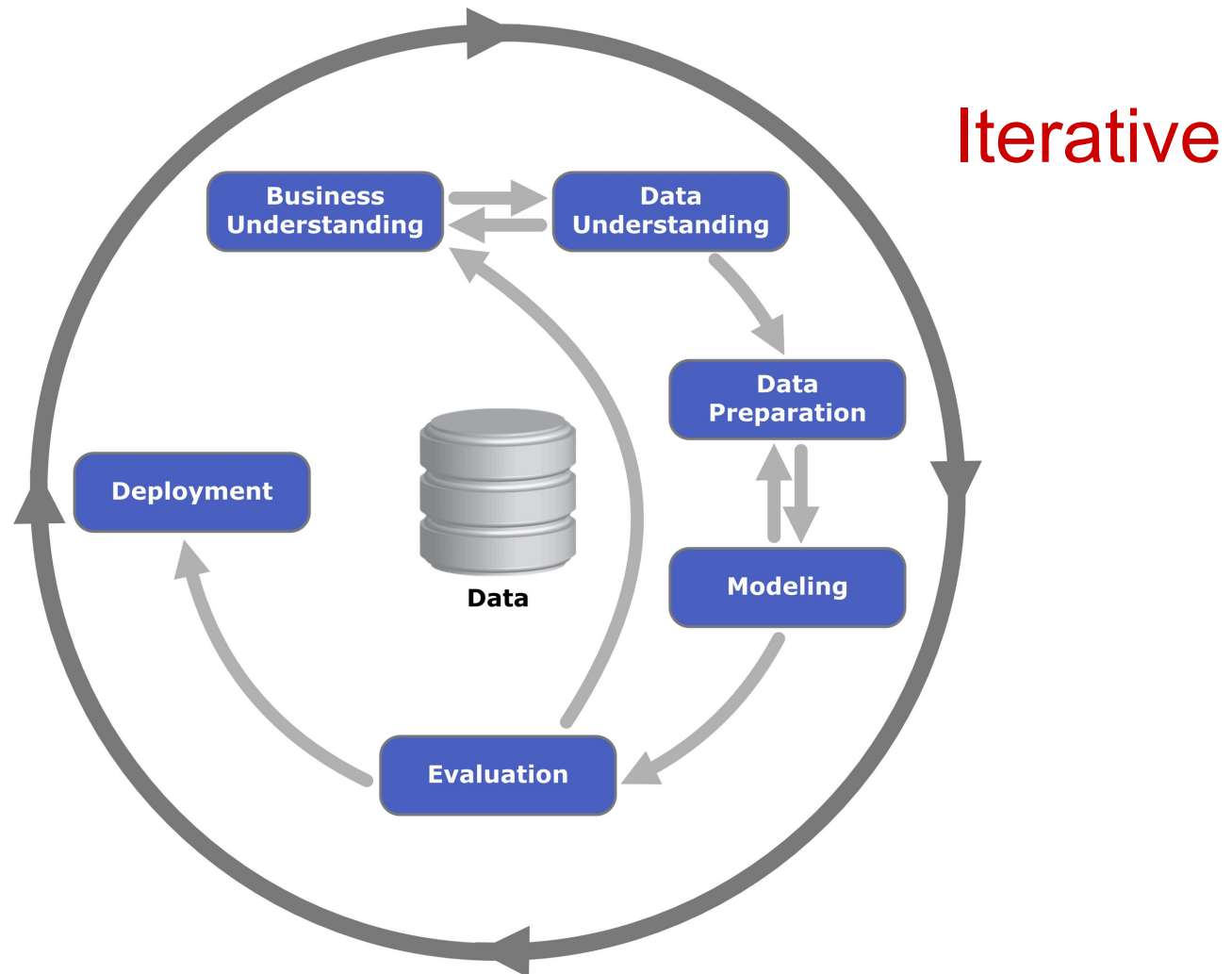
# PHASE 6: DEPLOYMENT

- Review and document project
  - Summarize findings and recommend uses
- Deploy model
  - Optimize model for inferencing
  - Integrate model into decision-making process
  - Package model
  - Make model available
- Create plan for model monitoring & maintenance
  - Monitoring model performance
  - Plan for updating model





# MACHINE LEARNING PROCESS



# BIG DATA ANALYTICS

- Machine Learning Overview
- **Data Exploration**
- Data Preparation
- Modeling
- Spark MLlib
- Assignments

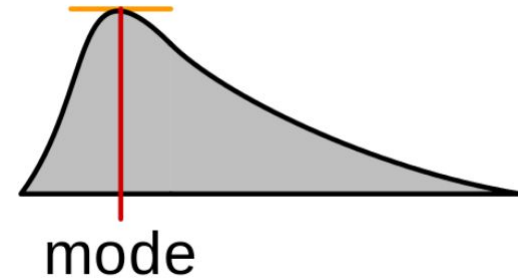
# DATA EXPLORATION

- Definition
  - Preliminary investigation of your data
- Purpose
  - To gain better understanding of specific characteristics of the data
  - To look for: Correlations, general trends, outliers, etc.
- Also referred to as 'EDA'
  - Exploratory Data Analysis
- Techniques
  - Data validation
  - Summary statistics
  - Visualization

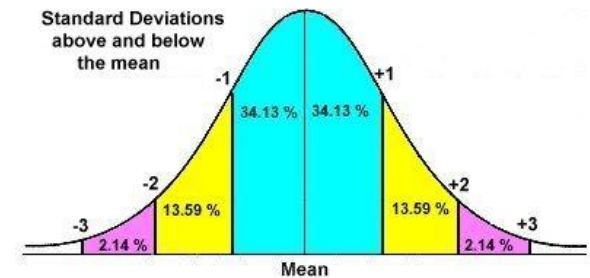
# DATA VALIDATION

- Check dimensions
  - Do number of rows/columns match what's expected in dataset?
- Check values
  - Should temperature values be in F or C?
  - Is data type of Date column date or timestamp?
- Check missing values
  - How many samples have missing values?
  - Which columns have mostly missing values?

# SUMMARY STATISTICS



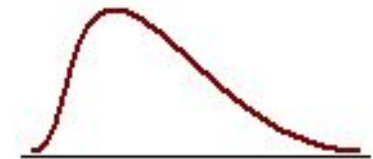
- Summarize & describe a set of data values
- Location: mean, median, mode
- Spread: min, max, range, standard deviation
- Shape: skewness, kurtosis
- Dependence



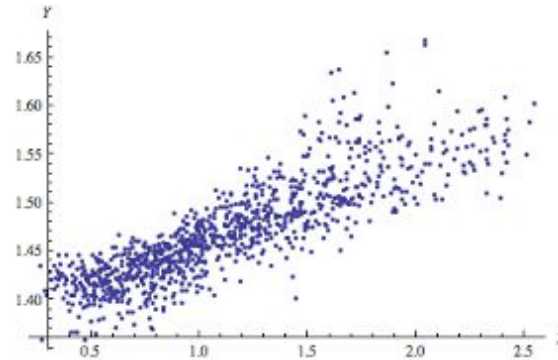
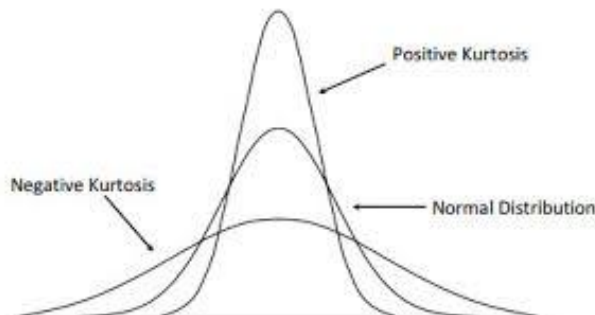
Negatively skewed distribution  
or Skewed to the left  
Skewness  $< 0$



Normal distribution  
Symmetrical  
Skewness  $= 0$



Positively skewed distribution  
or Skewed to the right  
Skewness  $> 0$

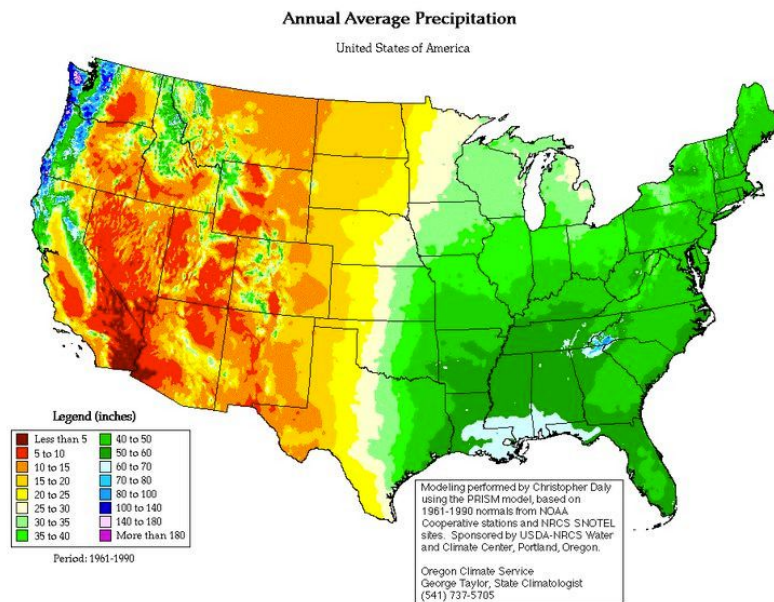
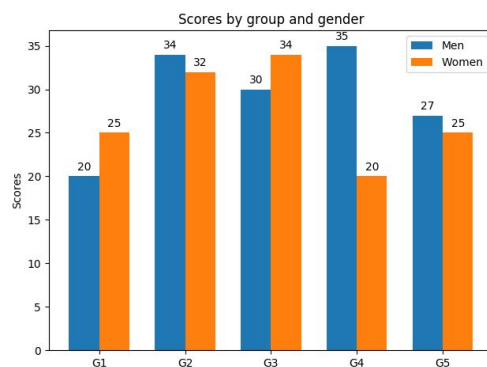
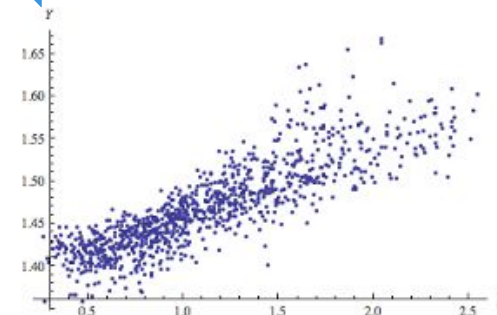


# DATA VISUALIZATION

- Intuitive way to look at data
- Useful for communicating results
- Should be used with summary statistics for EDA

- Examples

- Histogram
- Line plot
- Bar plot
- Box plot
- Scatter plot
- Heat map



# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- **Data Preparation**
- Modeling
- Spark MLlib
- Assignments

# DATA PREPARATION

- Goal
  - Create data for analysis
- Other names
  - Data preprocessing
  - Data munging
  - Data wrangling
- Activities
  - Clean data
    - Identify and address data quality issues
  - Feature engineering
    - Select features to use
    - Transform features



# DATA CLEANING

- Data quality issues

- Missing values
- Duplicate data
- Inconsistent data
- Noise
- Outliers

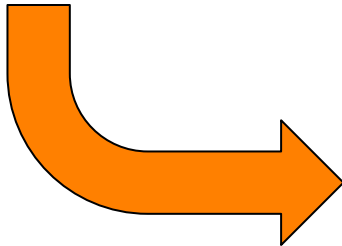


- Addressing data quality issues

- Remove data with missing values
- Merge duplicate records
- Generate best estimate for invalid values

# REMOVE MISSING DATA

Name	Age	Income
Angela	34	80
Sidney	--	56
Ratan	10	--
Kiril	68	--
Zhou	45	120

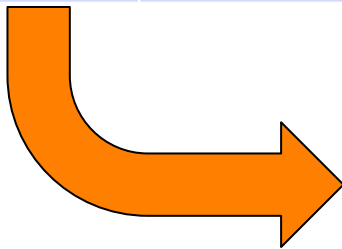


Name	Age	Income
Angela	34	80
<del>Sidney</del>	<del>--</del>	<del>56</del>
<del>Ratan</del>	<del>10</del>	<del>--</del>
<del>Kiril</del>	<del>68</del>	<del>--</del>
Zhou	45	120

Drop rows with any or all null values

# IMPUTING MISSING DATA

Name	Age	Income
Angela	34	80
Sidney	--	56
Ratan	10	--
Kiril	68	--
Zhou	45	120



Name	Age	Income
Angela	34	80
Sidney	50	56
Ratan	10	50
Kiril	68	50
Zhou	45	120

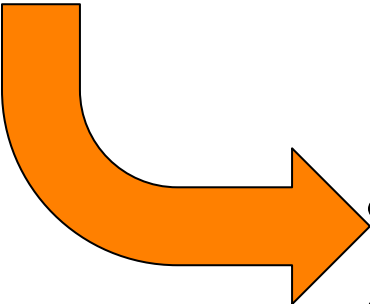
Replace missing values with something reasonable

- Mean
- Median
- Most frequent
- Sensible value based on application

# DUPLICATE DATA

Name	Address
Sidney	7800 West View Street
Sid	7800 West View Street
Kiril	45 East 5 <sup>th</sup> St
Kiril	1220 Mill Avenue

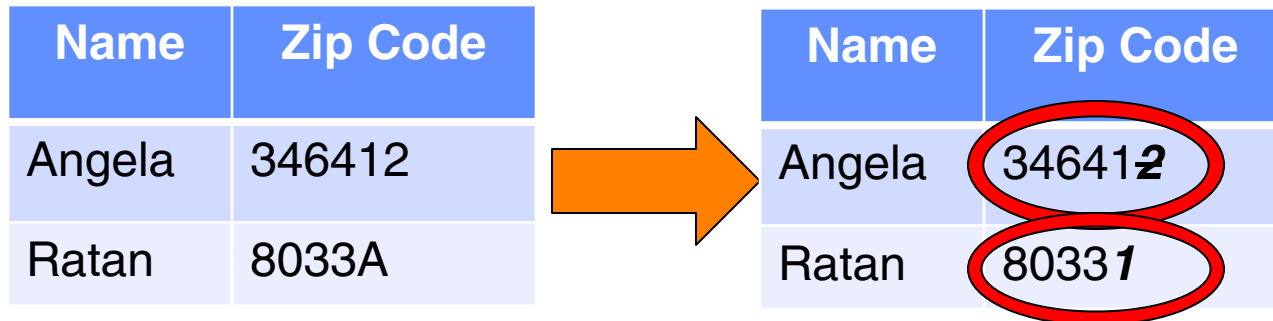
- Delete older record.
- Merge duplicate records



Name	Address
Sidney	7800 West View Street
<del>Sid</del>	<del>7800 West View Street</del>
<del>Kiril</del>	<del>45 East 5<sup>th</sup> St</del>
Kiril	1220 Mill Avenue

# INVALID DATA

- Use external data source to get correct value
- Apply reasoning and domain knowledge to come up with reasonable value.



The diagram illustrates the process of identifying invalid data. It shows a transformation from a valid dataset to one containing errors. An orange arrow points from the left table to the right table.

Name	Zip Code
Angela	346412
Ratan	8033A

Name	Zip Code
Angela	34641 <del>2</del>
Ratan	8033 <del>1</del>

- Note:  
Addressing data quality issues requires  
**domain knowledge!**

# DATA PREPARATION

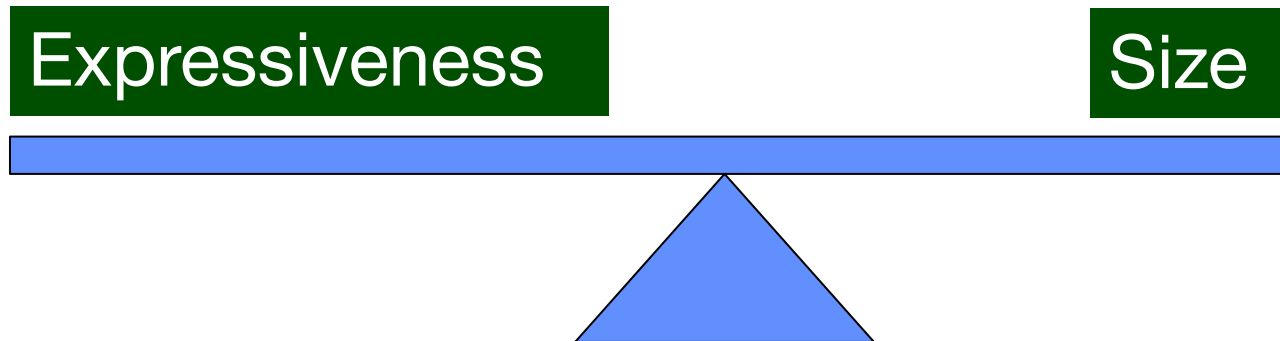
- Goal
  - Create data for analysis
- Other names
  - Data preprocessing
  - Data munging
  - Data wrangling
- Activities
  - Clean data
    - ▢ Identify and address data quality issues
  - Feature engineering
    - ▢ Select features to use
    - ▢ Apply transformations to data

# FEATURE ENGINEERING

- Feature selection
  - Combining features
  - Adding or removing features
- Feature transformation
  - Scaling
  - Aggregation
  - Discretization
  - One-hot encoding
  - Dimensionality reduction

# FEATURE SELECTION

- Goal
  - Characterize problem with smallest set of features
- Motivation
  - Simplify model
  - Faster to train model
  - Enhances model robustness

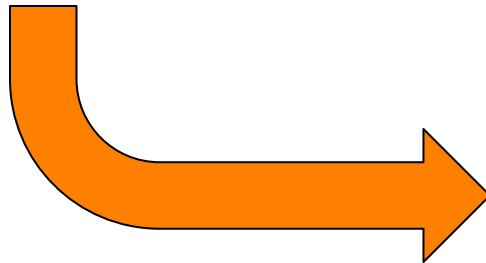




# ADDING FEATURES

New features derived from existing features

Name	State
Angela	AK
Sidney	CA
Ratan	WA
Kiril	OR
Zhou	CA



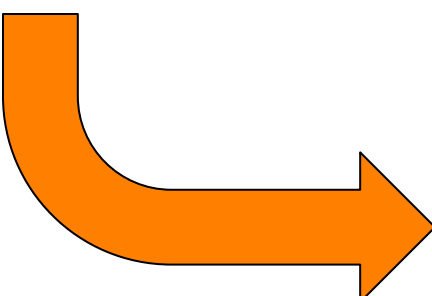
Name	State	<i>In-State</i>
Angela	AK	<b><i>F</i></b>
Sidney	CA	<b><i>T</i></b>
Ratan	WA	<b><i>F</i></b>
Kiril	OR	<b><i>F</i></b>
Zhou	CA	<b><i>T</i></b>

# REMOVING FEATURES

- Features with a lot of missing values
- Features that are very correlated with another feature
- Features that do not have predictive power
  - Based on correlation or information gain with target variable
- Irrelevant features
  - ID, row number, etc.

# COMBINING FEATURES

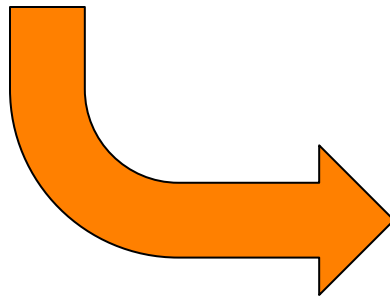
Name	Height	Weight
Angela	1.8	68
Sidney	1.5	70
Ratan	2.0	84
Kiril	1.3	54
Zhou	2.0	61



Name	Height	Weight	<i><b>BMI</b></i>
Angela	180	68	<b>21</b>
Sidney	153	70	<b>30</b>
Ratan	204	84	<b>20</b>
Kiril	133	44	<b>25</b>
Zhou	208	81	<b>19</b>

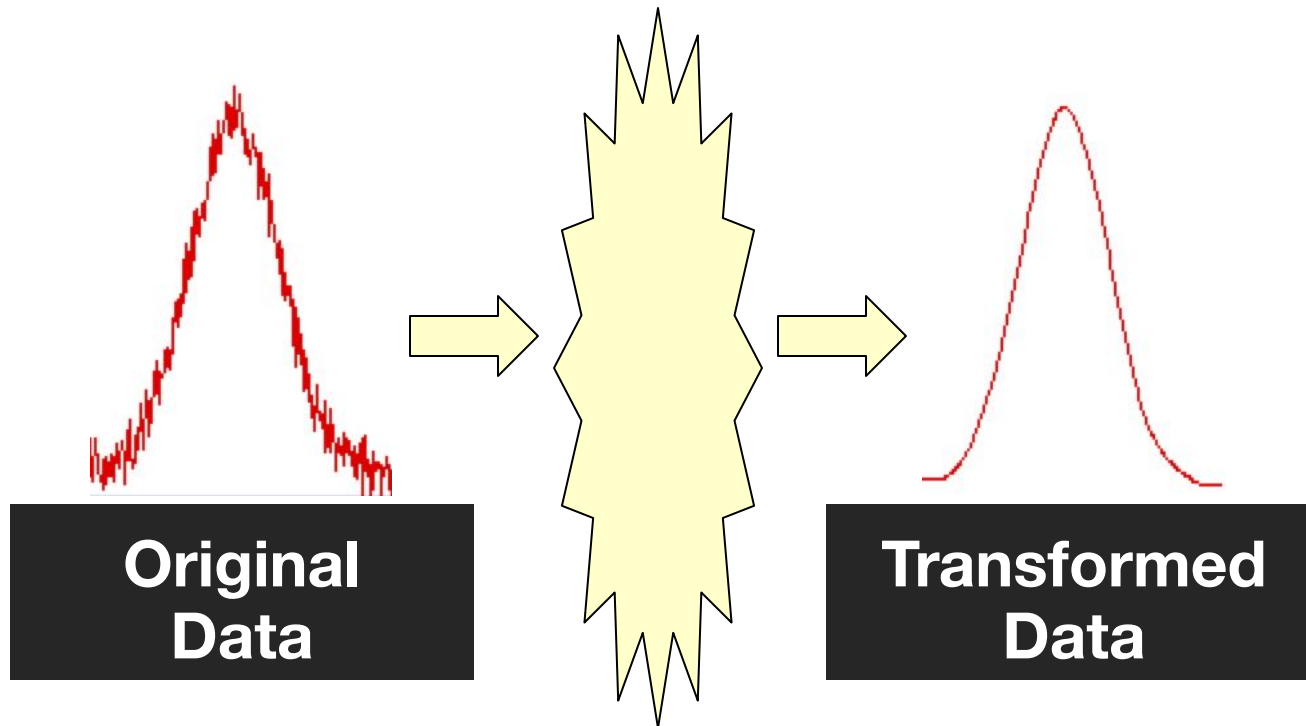
# BREAKING UP FEATURES

Address
430 Park Drive, CA, 97283
7800 W. View Street, FL, 34642
1243 Mountain Ave., CO, 80334
1220 Mill Avenue, IL, 54622
4345 Apple Lane, WA, 98421

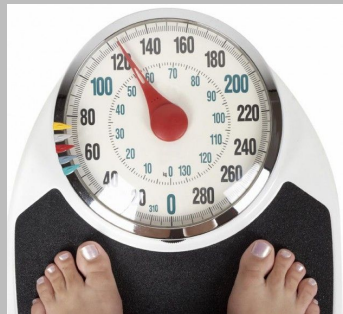


Address	State	Zip
430 Park Drive	CA	97283
7800 W. View Street	FL	34642
1243 Mountain Ave.	CO	80334
1220 Mill Avenue	IL	54622
4345 Apple Lane	WA	98421

# FEATURE TRANSFORMATION



# SCALING



**Weight**

**Scaled Values**



**Height**

# WAYS TO SCALE

- **Mean center**

$$x_{new} = x - \text{mean}(x)$$

- **Min-Max Normalization**

- Rescale to [0,1]

$$x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Rescale to [a,b]

$$x_{new} = a + \frac{(x - \min(x)) (b - a)}{\max(x) - \min(x)}$$

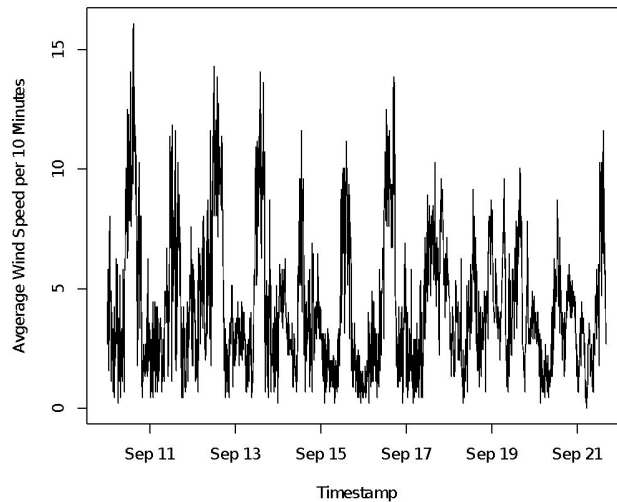
- **Standardization**

(Z-score Normalization)

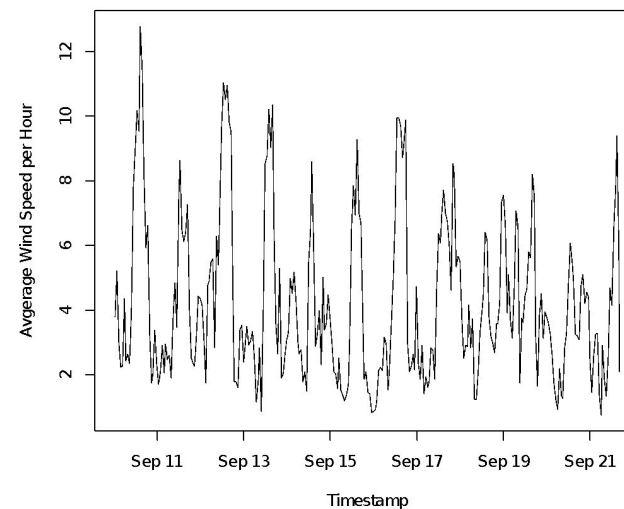
$$x_{new} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

# AGGREGATION

Avg Wind Speed  
(every 10 minutes)



Avg Wind Speed  
(every 60 minutes)





# RECODING FEATURES

- **Discretization**

- Reformat continuous feature as categorical
- Example
  - Customer's age => {teenager, young adult, adult, senior}

- **One-hot encoding**

- Reformat categorical feature as vector
- To avoid imposing ordering between categories
- Example
  - Integer encoding: red=0, green=1, blue=2
  - One-hot encoding
    - ❖ Red: {1,0,0}
    - ❖ Green: {0,1,0}
    - ❖ Blue: {0,0,1}

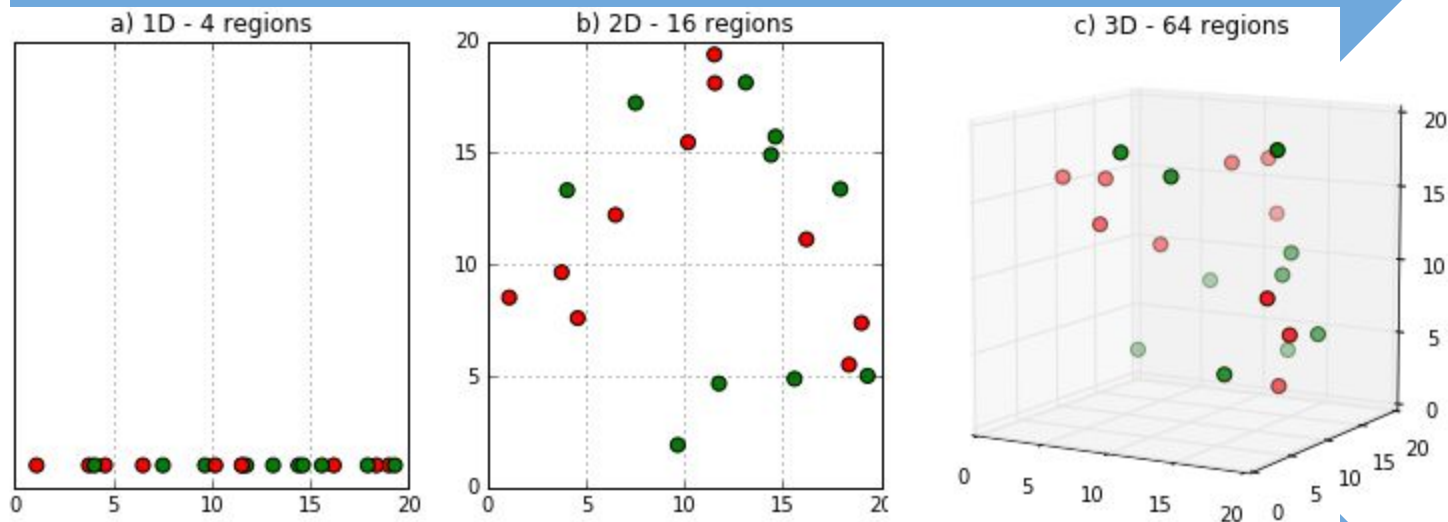
# DIMENSIONALITY REDUCTION

- **Goal**
  - Transform data in high-dimensional space to a space with fewer dimensions
- **Motivation**
  - Reduces computational and storage requirements to process data and train model
  - Reduces complexity (number of parameters) of model
  - Enhances model's performance
  - Addresses issues with curse of dimensionality

# CURSE OF DIMENSIONALITY

As dimensionality increases, feature space grows exponentially

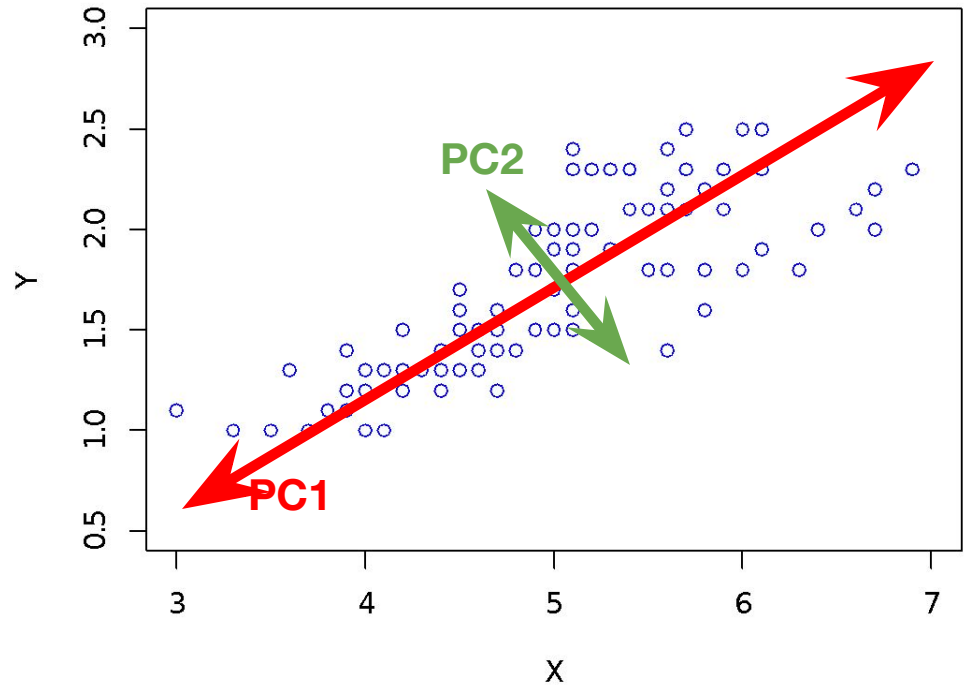
**Data gets increasingly sparse**



**Analysis results degrade**

# PRINCIPAL COMPONENT ANALYSIS (PCA)

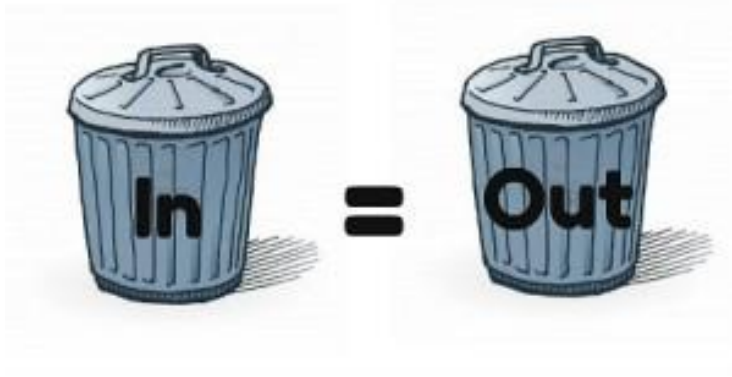
- Map data from high to low-dimensional space
- Want to capture as much of variation in data as possible
- Principal components (PCs) capture directions with most amount of variance
- Data is projected to new dimensions defined by PCs
- PCs are orthogonal to each other



# DATA PREPARATION

- Data Cleaning
- Feature Engineering

**Always Remember!**



Data preparation is very important for meaningful analysis

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - Categories of Machine Learning Techniques
  - Building and Applying a Model
  - Classification
  - Regression
  - Cluster Analysis
- Spark MLlib
- Assignments

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - **Categories of Machine Learning Techniques**
  - Building and Applying a Model
  - Classification
  - Regression
  - Cluster Analysis
- Spark MLlib
- Assignments

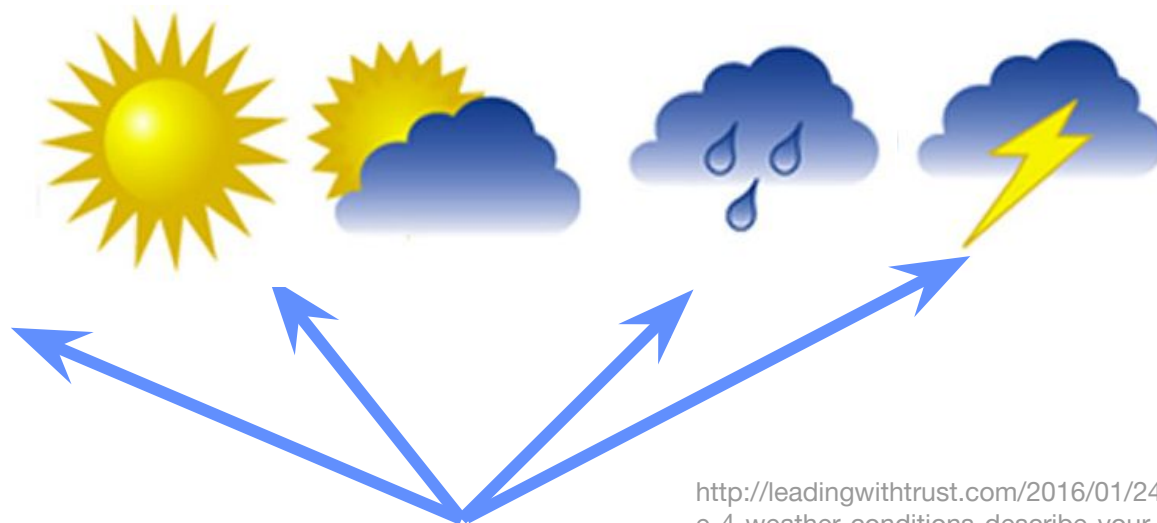
# MAIN CATEGORIES OF MACHINE LEARNING TECHNIQUES

- Classification
- Regression
- Cluster Analysis



# CLASSIFICATION

- Goal: Predict category given input data
  - Target is categorical variable



<http://leadingwithtrust.com/2016/01/24/which-of-these-4-weather-conditions-describe-your-leadership/>

- Examples
  - Classify tumor as benign or malignant
  - Determine if credit card transaction is legitimate or fraudulent
  - Identify customer as residential, commercial, public
  - Predict if weather will be sunny, cloudy, windy, or rainy

# REGRESSION

- Goal: Predict numeric value given input data
  - Target is numeric variable



[www.wallstreetpoint.com](http://www.wallstreetpoint.com)

- Examples
  - Predict price of stock
  - Estimate demand for a product based on time of year
  - Determine risk of loan application
  - Predict amount of rain

# CLUSTER ANALYSIS

- Goal: Organize similar items into groups



<http://www.bostonlogic.com/blog/2014/01/segment-your-leads-to-get-better-results/>

- Examples
  - Group customer base into segments for effective targeted marketing
  - Identify areas of similar topography (desert, grass, etc.)
  - Categorize different types of tissues from medical images
  - Discover crime hot spots

# SUPERVISED VS. UNSUPERVISED

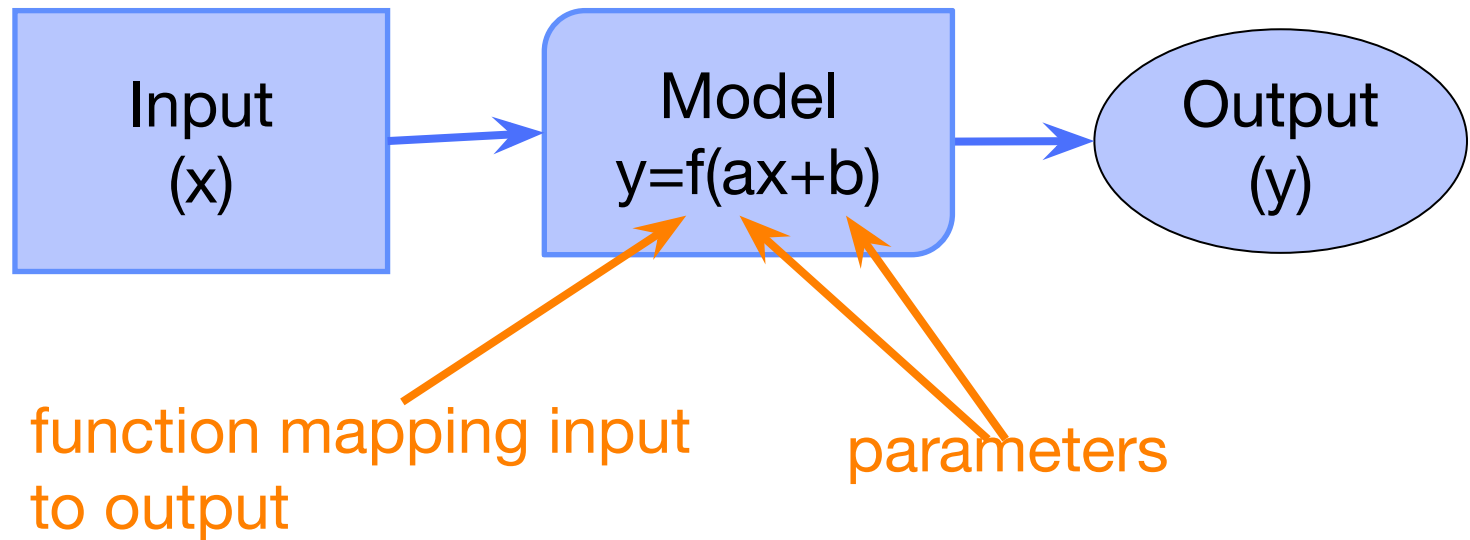
- Supervised Learning
  - Target (what you're trying to predict) is provided
    - 'Labeled' data
  - Classification and regression are supervised
- Unsupervised Learning
  - Target is unknown or unavailable
    - 'Unlabeled' data
  - Cluster analysis is unsupervised

# BIG DATA ANALYTICS

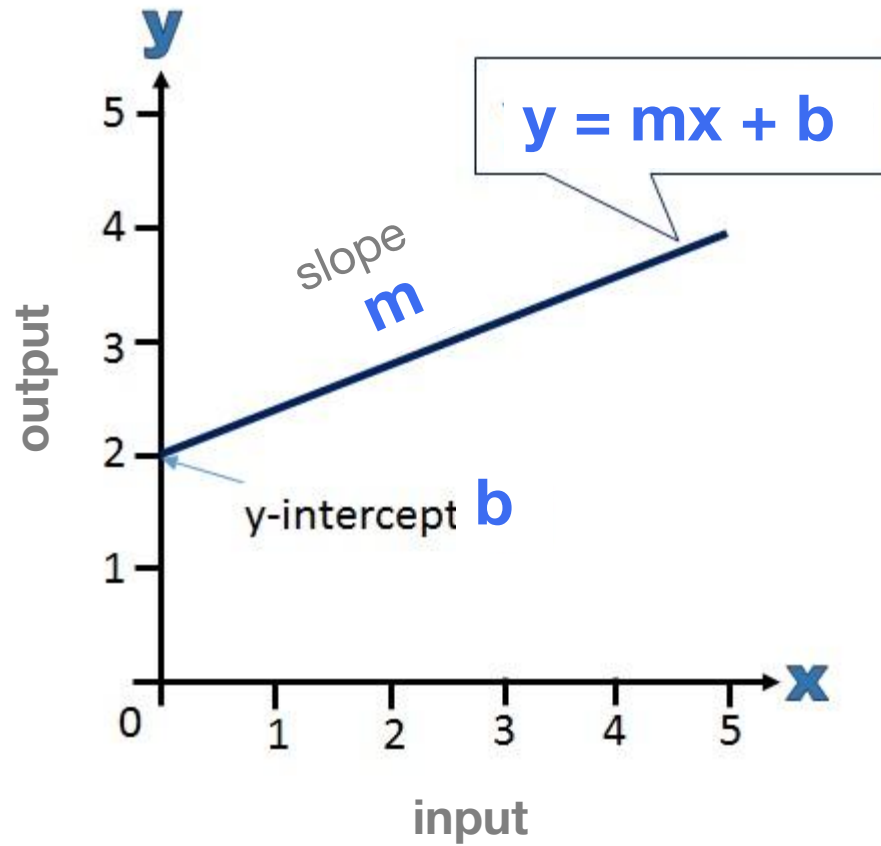
- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - Categories of Machine Learning Techniques
  - **Building and Applying a Model**
  - Classification
  - Regression
  - Cluster Analysis
- Spark MLlib
- Assignments

# MACHINE LEARNING MODEL

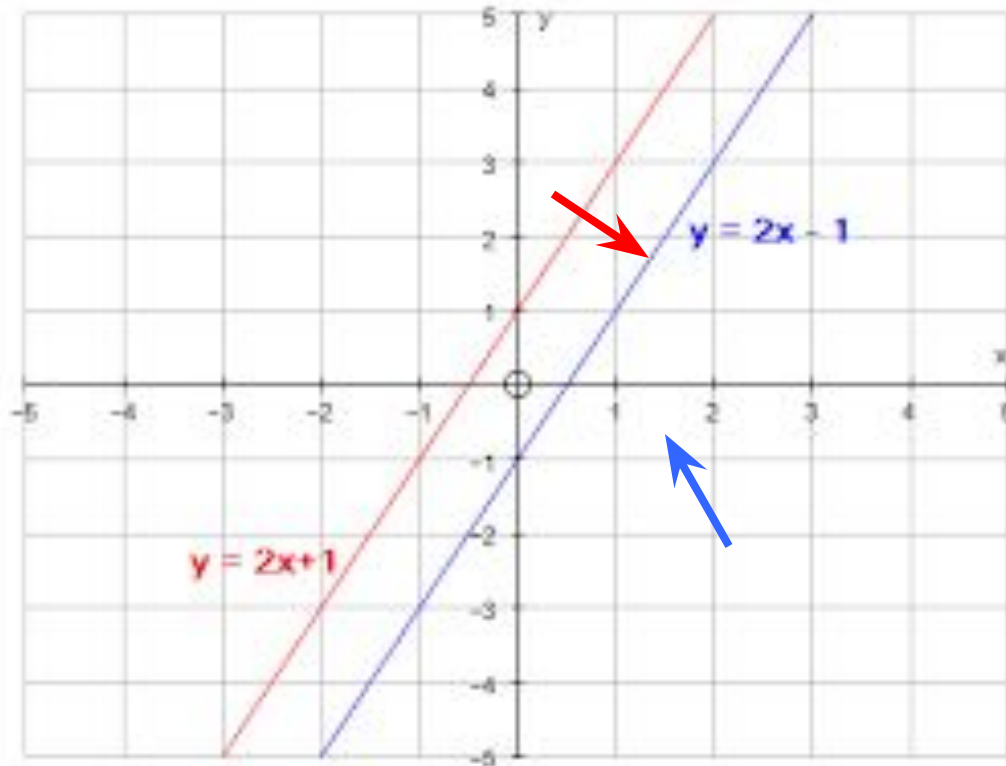
A mathematical model with parameters that map input to output



# EXAMPLE OF A MODEL



# ADJUSTING MODEL PARAMETERS



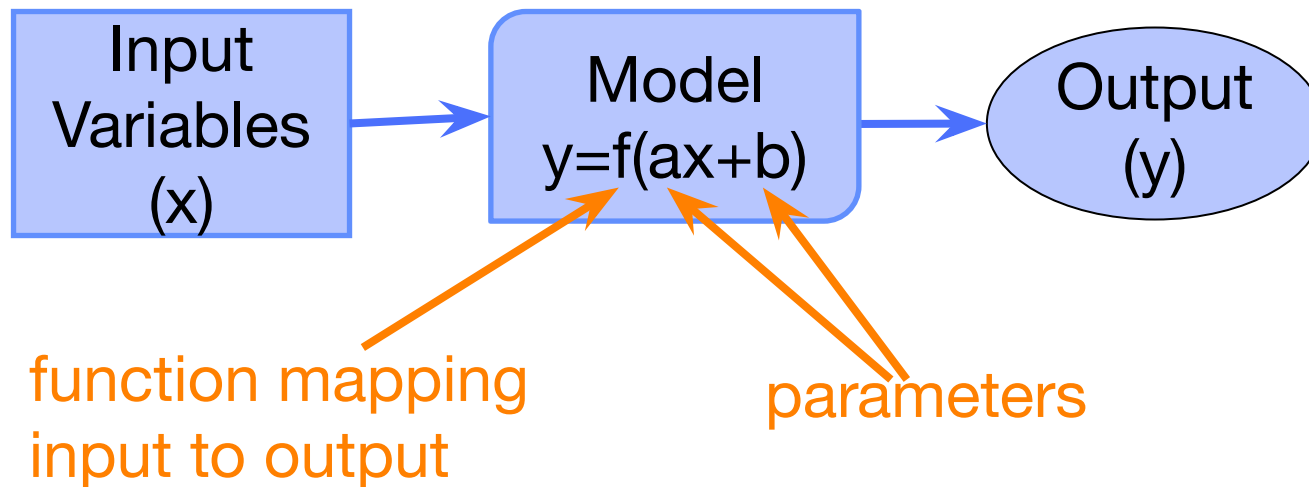
slope  $m = 2$   
y-intercept  $b = -1$   
 $x=1 \Rightarrow y=2*1-1=1$

slope  $m = 2$   
y-intercept  $b = +1$   
 $x=1 \Rightarrow y=2*1+1=3$

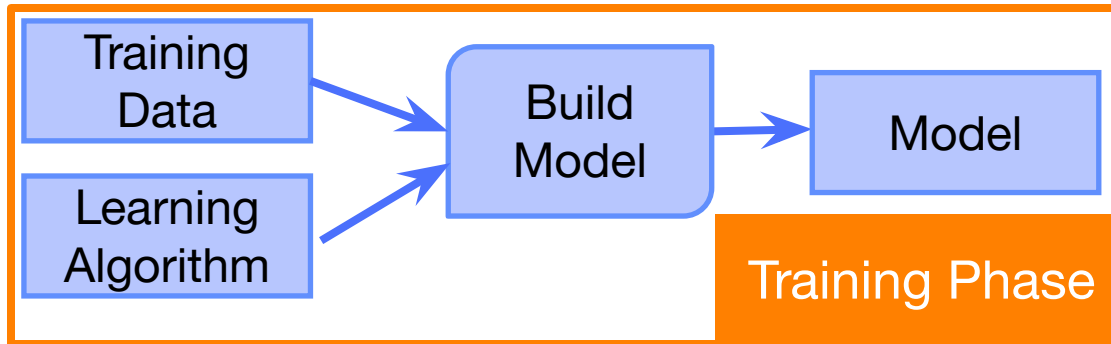


# BUILDING MACHINE LEARNING MODEL

- Model parameters are adjusted during model training to change input-output mapping
- Parameters are learned or estimated from data
  - “fitting the model”, “training the model”, “building the model”
- Goal: Minimize some error function

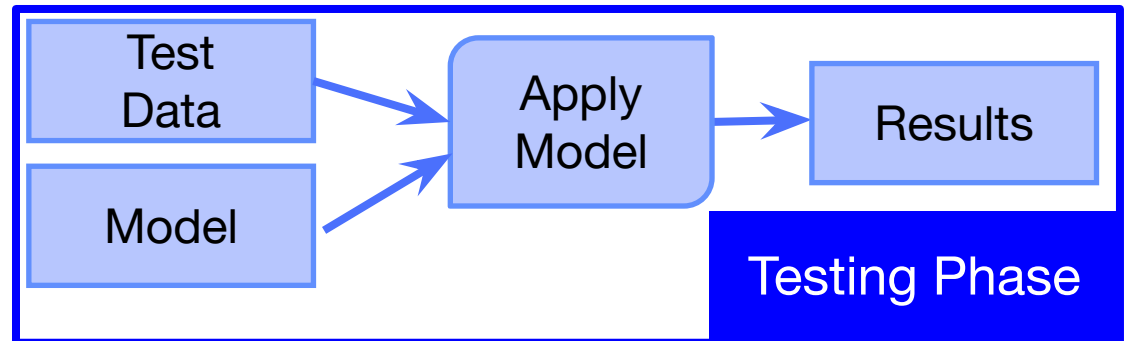


# BUILDING VS APPLYING MODEL

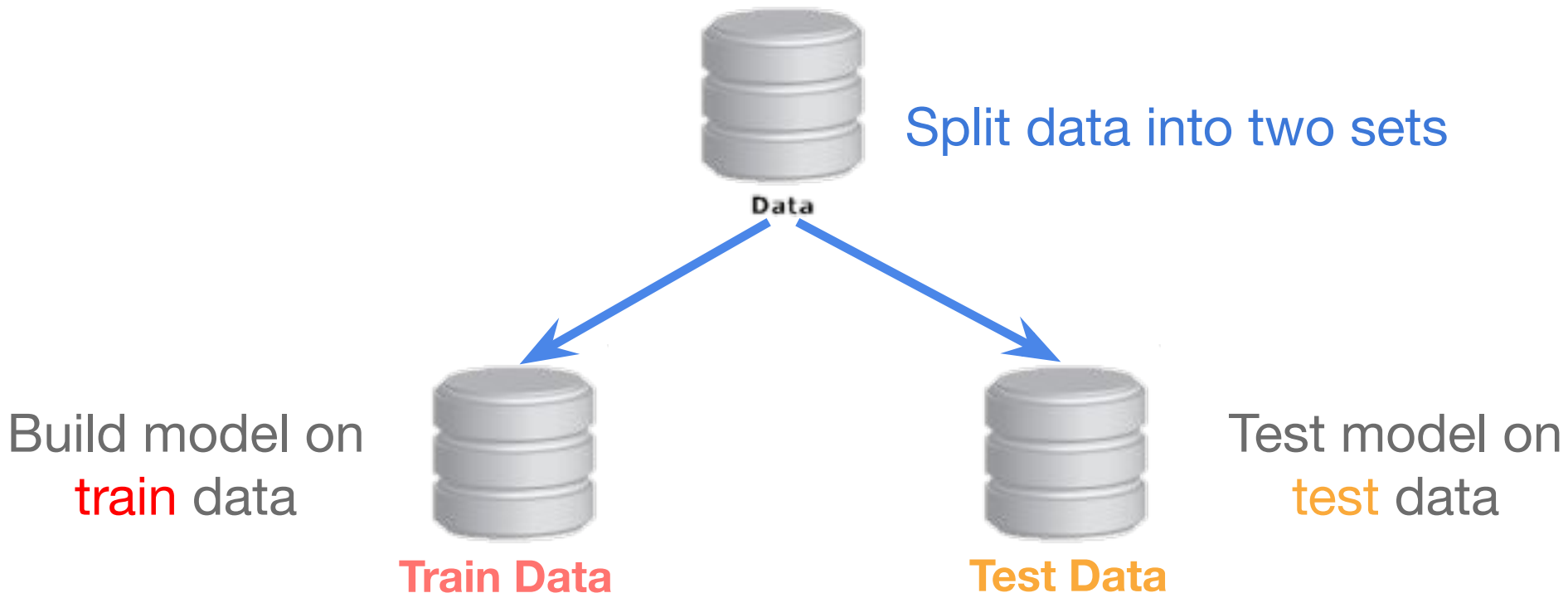


Adjust model  
parameters  
“Train”

Test model on  
new data  
“Inference”



# GENERALIZATION



Goal: Want model to perform well on data it was not trained on, i.e., to **generalize** well to unseen data

# OVERFITTING & GENERALIZATION

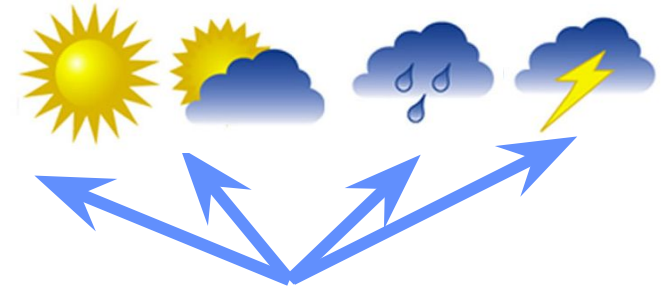
- Overfitting
  - Model is fitting to noise in data instead of to underlying distribution of data
- Reasons for overfitting
  - Training set is too small
  - Model is too complex, i.e., has too many parameters
- Overfitting leads to poor generalization
  - Model that overfits will not generalize well to new data

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - Categories of Machine Learning Techniques
  - Building and Applying a Model
  - **Classification**
  - Regression
  - Cluster Analysis
- Spark MLlib
- Assignments

# CLASSIFICATION

- Goal: Predict category given input data
  - Target is categorical variable



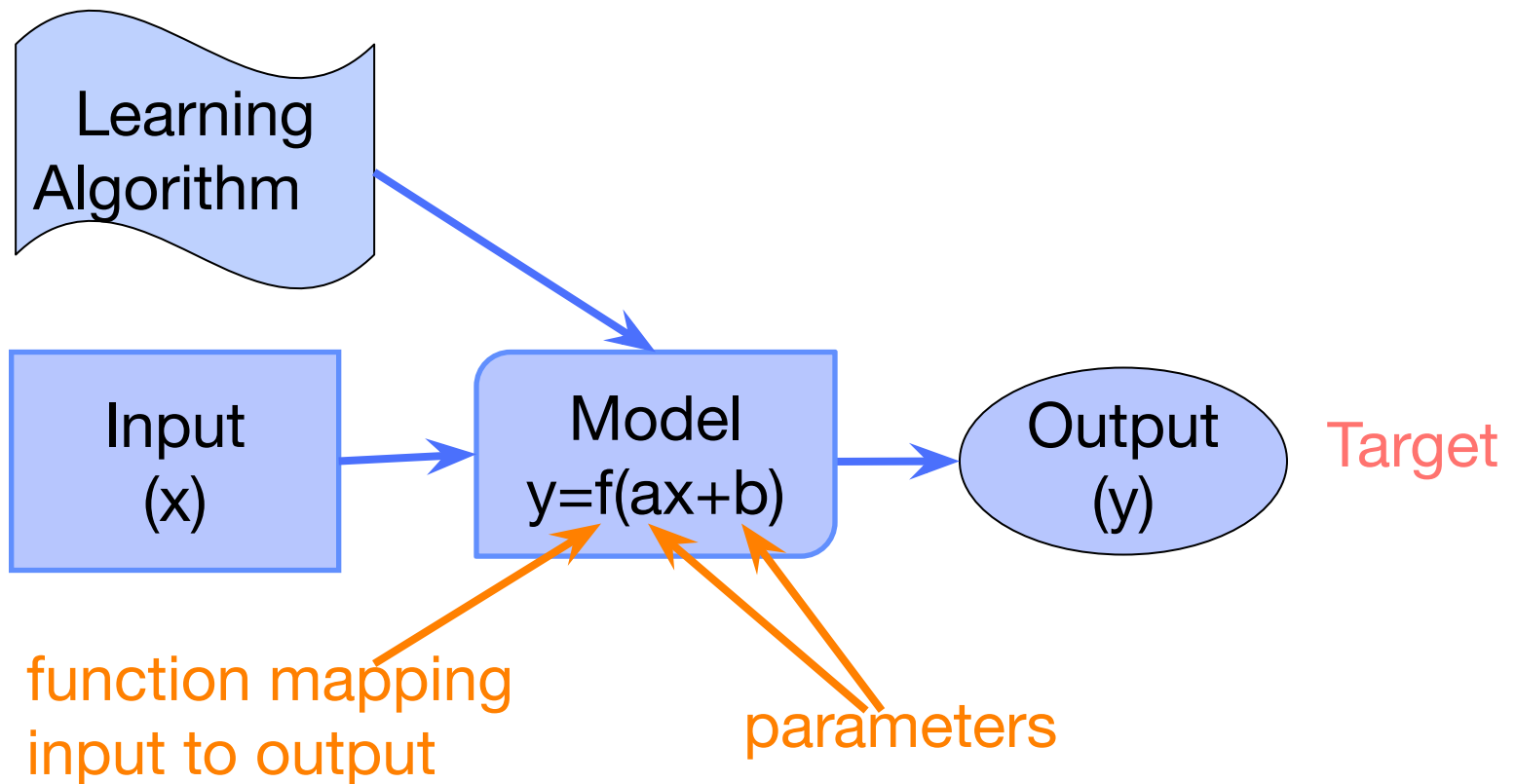
**Input Variables**

**Target Variable**

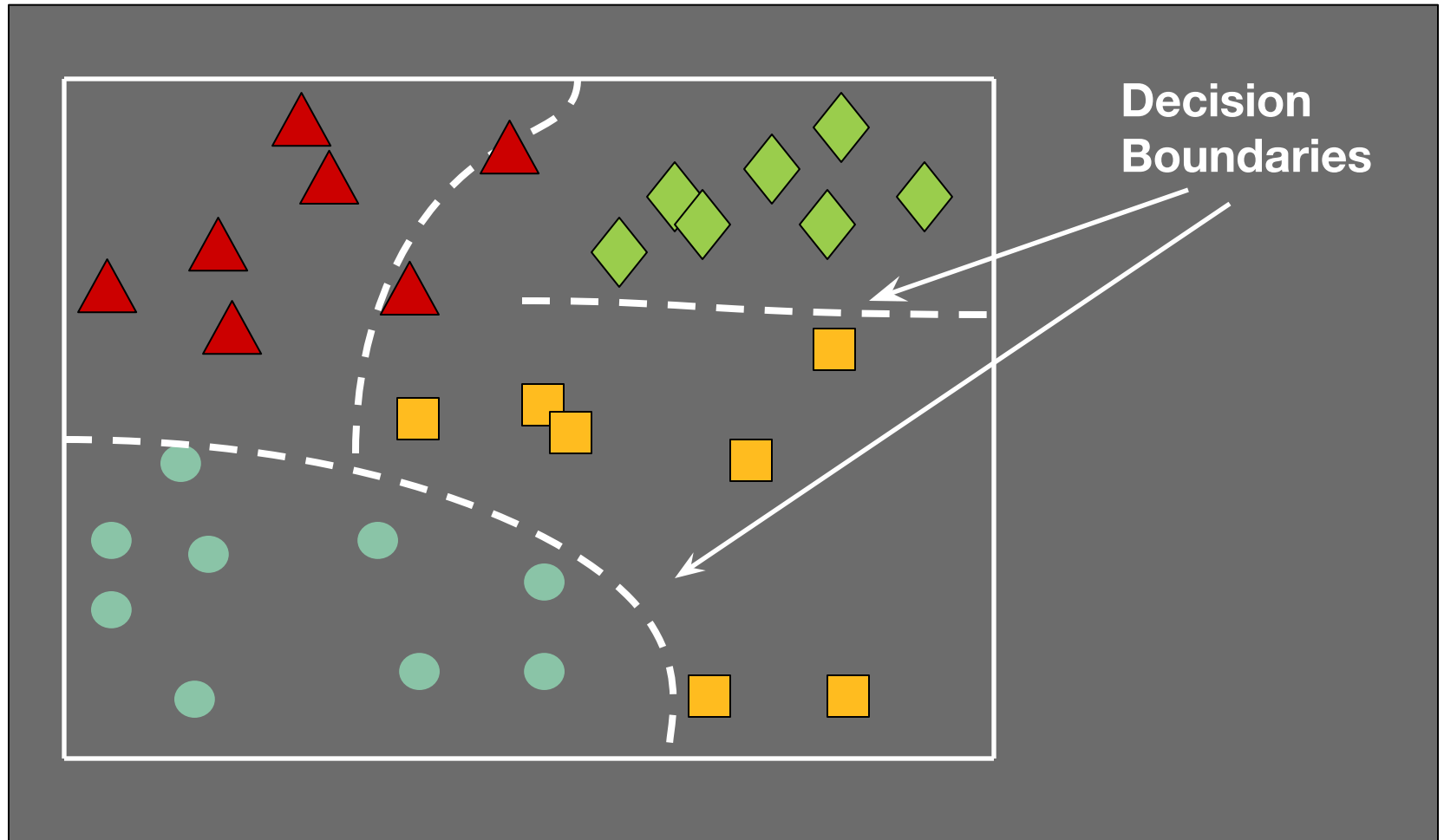
Temperature	Humidity	Wind Speed	Weather
79	48	2.7	Sunny
60	80	3.8	Rainy
68	45	17.9	Windy
57	77	4.2	Cloudy

# BUILDING CLASSIFICATION MODEL

- Want to get model outputs to match targets
- Learning algorithm used to adjust model parameters to minimize difference between outputs and targets



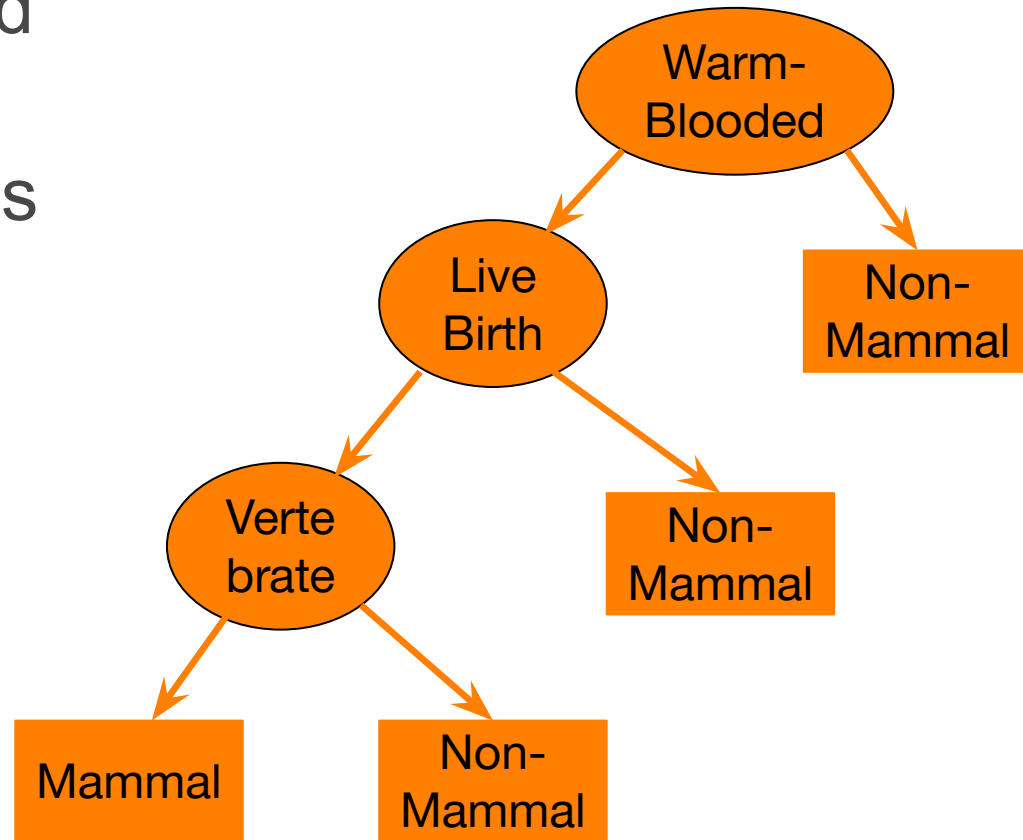
# BUILDING CLASSIFICATION MODEL





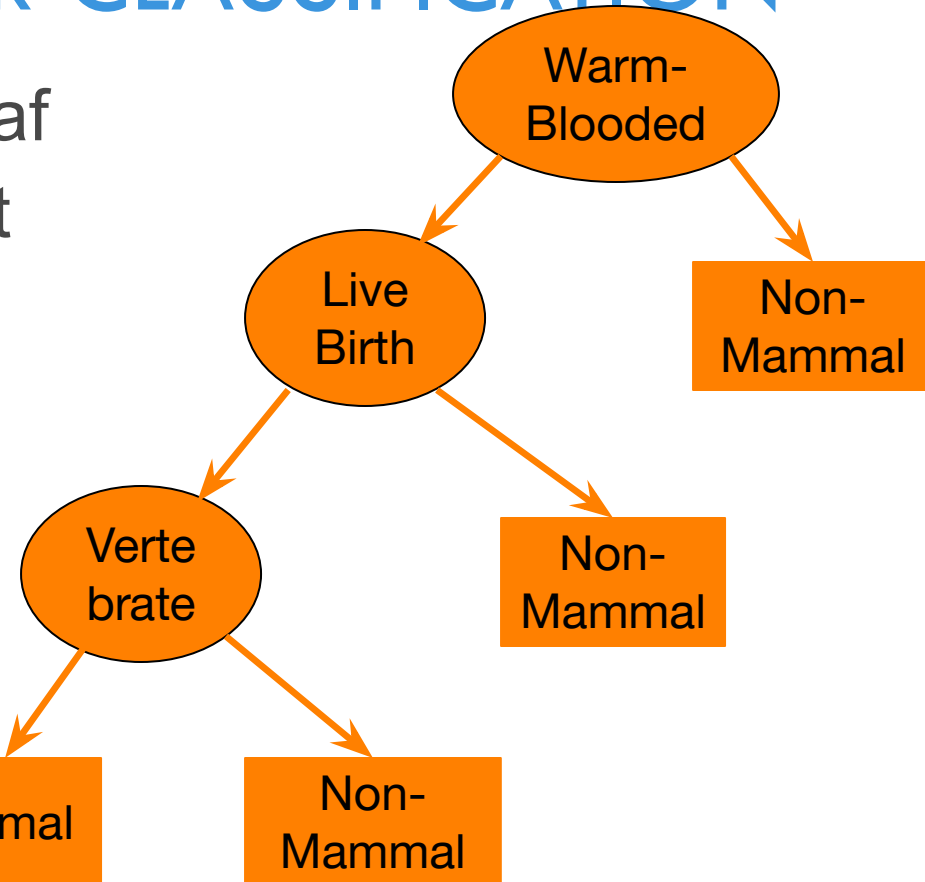
# DECISION TREE FOR CLASSIFICATION

- Hierarchical structure with nodes and directed edges
- Root and internal nodes have test conditions
- Leaf nodes have class labels
- Paths from root to leaf represent classification rules



# DECISION TREE FOR CLASSIFICATION

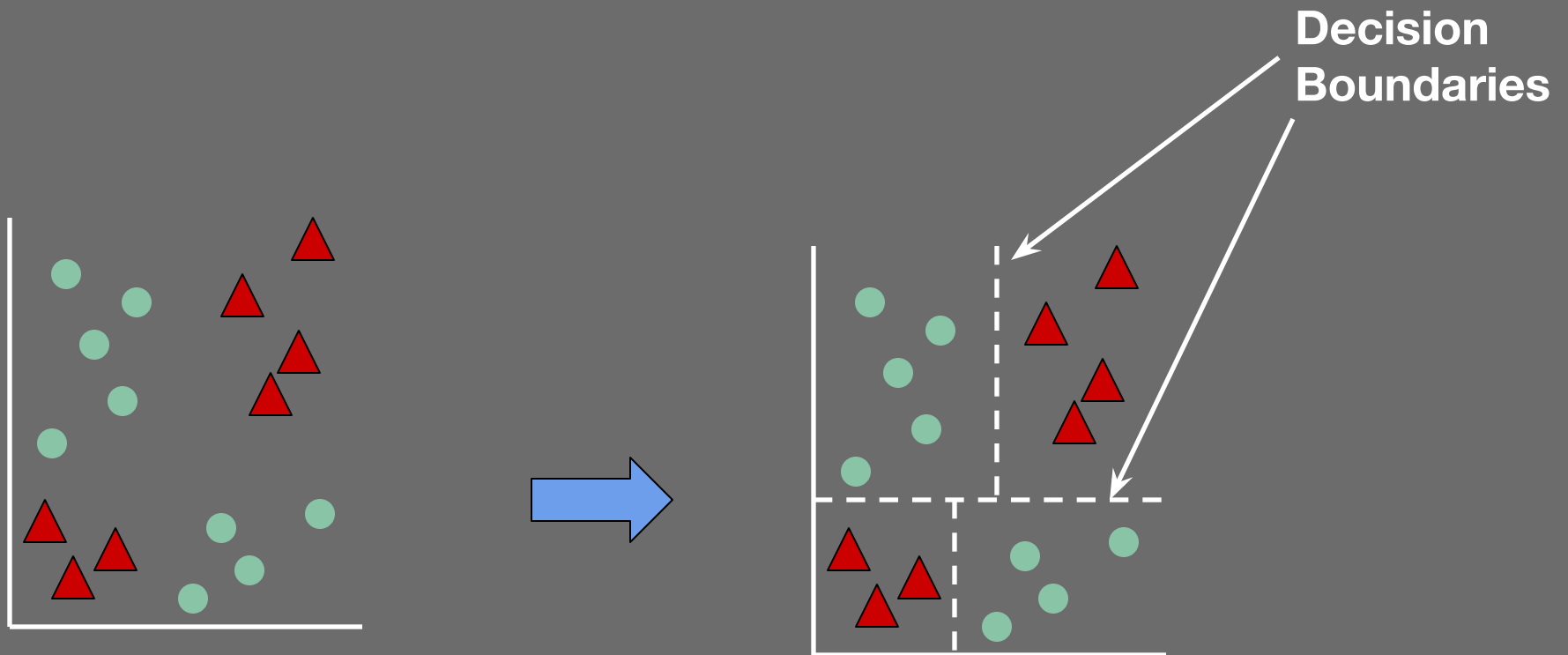
- Traverse tree from root to leaf
- At each node, answer to test condition determines child node to move to
- Category at leaf node determines label for sample



Warm-Blooded	Live Birth	Vertebrate	Target Label
Yes	Yes	Yes	Mammal

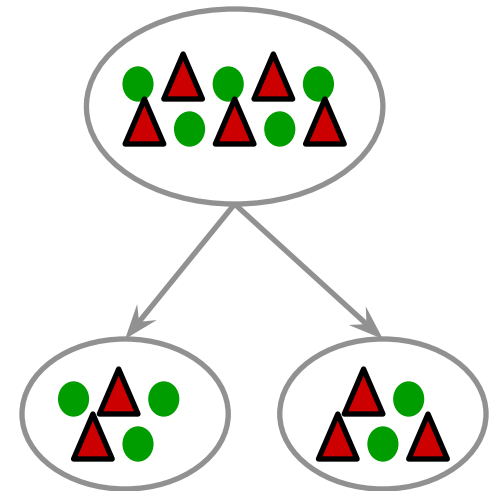
# CONSTRUCTING DECISION TREE

Idea: Split data into “pure” regions



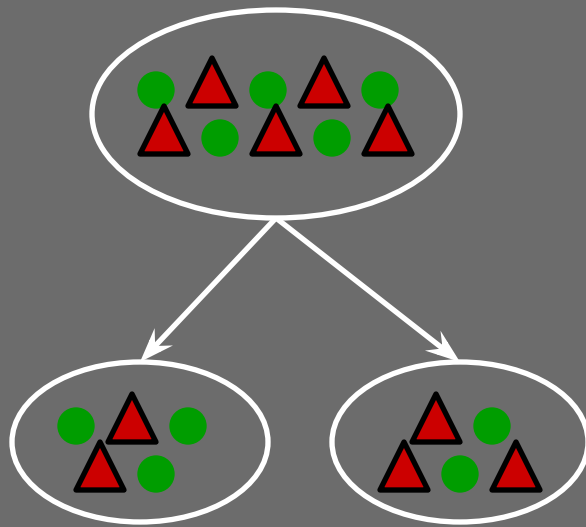
# CONSTRUCTING DECISION TREE

- Start with all samples at a node
- Partition samples based on input to create purest subsets
- Repeat to partition data into successively purer subsets
- Also referred to as **tree induction**

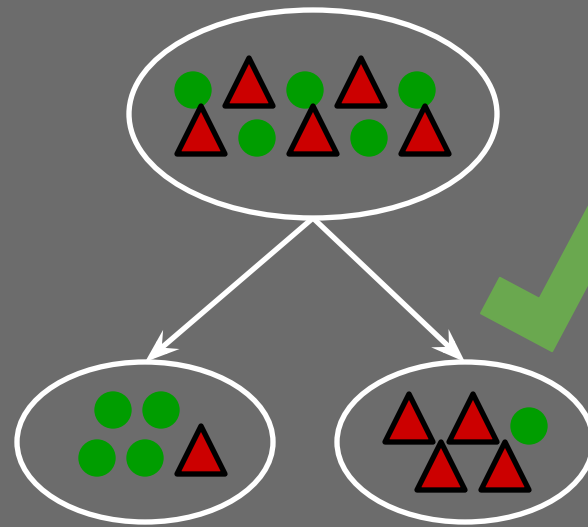


# HOW TO DETERMINE BEST SPLIT?

Want subsets to be as homogeneous as possible



**Less homogeneous =  
More pure**



**More homogeneous =  
More pure**

# IMPURITY MEASURE

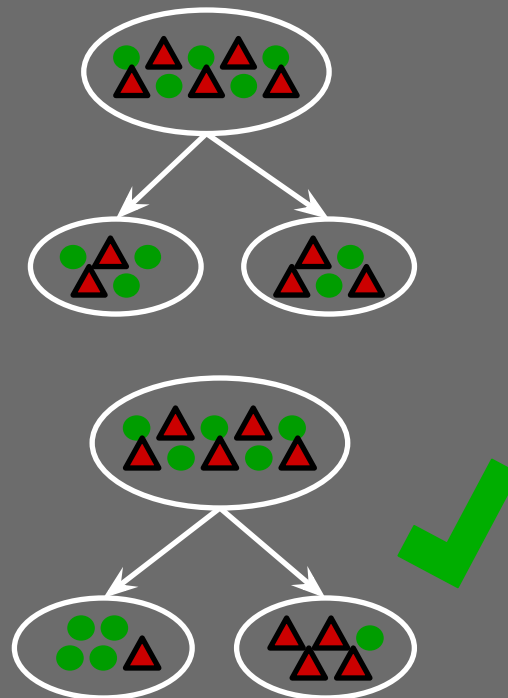
To compare different ways to split data in a node

Gini  
Index



Higher =  
Less pure

Lower =  
More pure



# GINI INDEX

- Gini Index for a given node  $t$ :

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$

where  $p(j | t)$  is the relative frequency of class  $j$  at node  $t$

- Gini Index measures impurity of a data partition
  - Maximum is  $(1 - 1/n_c)$ , when records are equally distributed among all classes, implying least interesting information.
  - Minimum is 0, when all records belong to one class, implying most interesting information.

# GINI INDEX FOR SPLIT

- When node  $p$  is split into  $k$  partitions, quality of split is computed as:

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

where  $n_i$  = number of records at child  $i$ ,  
 $n$  = number of records at node  $p$

- Take average of Gini scores of children, weighted by size of each child



# NODE SPLITTING CRITERIA

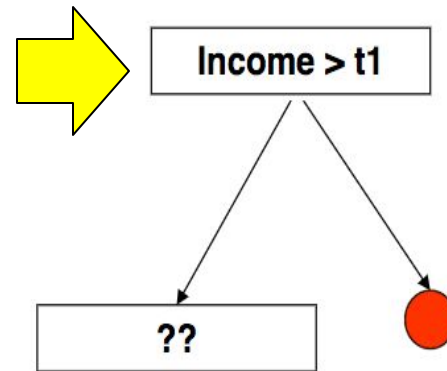
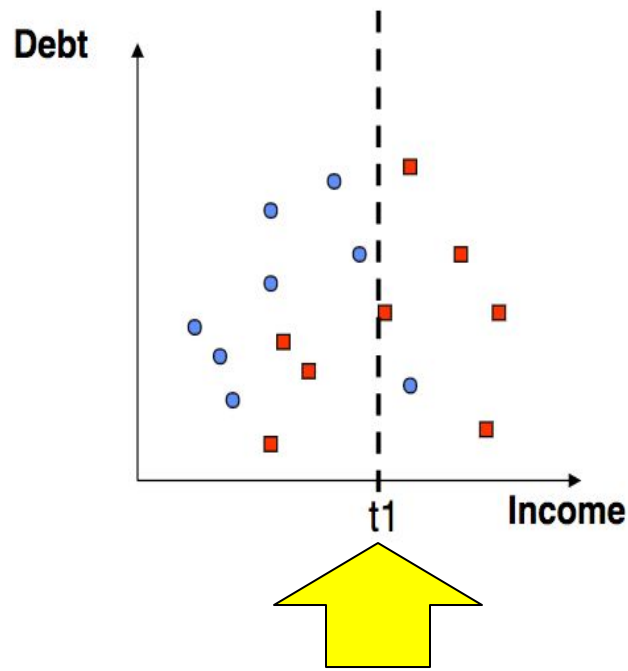
- Criteria to determine best split during tree induction:
  - Gini Index
  - Entropy (Information Gain)
  - Gain Ratio
  - Chi-Square
  - Minimum Description Length
  - Others

# WHEN TO STOP SPLITTING A NODE?

- All (or X% of) samples have same class label
- Number of samples in leaf reaches minimum
- Change in impurity measure is smaller than threshold
- Max tree depth is reached
- Others...

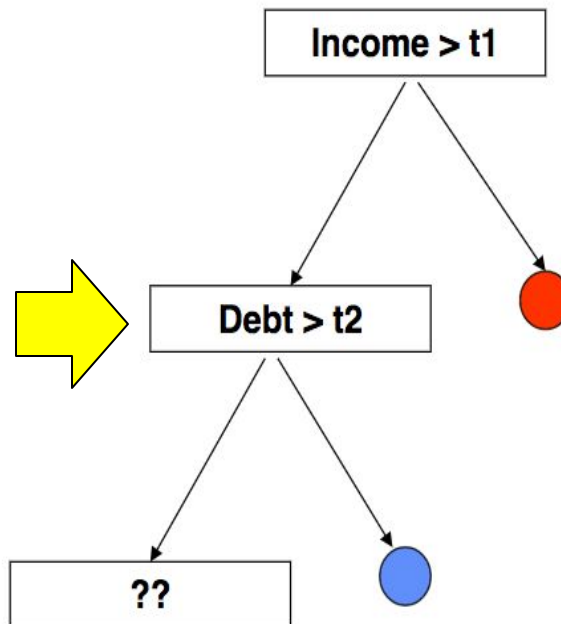
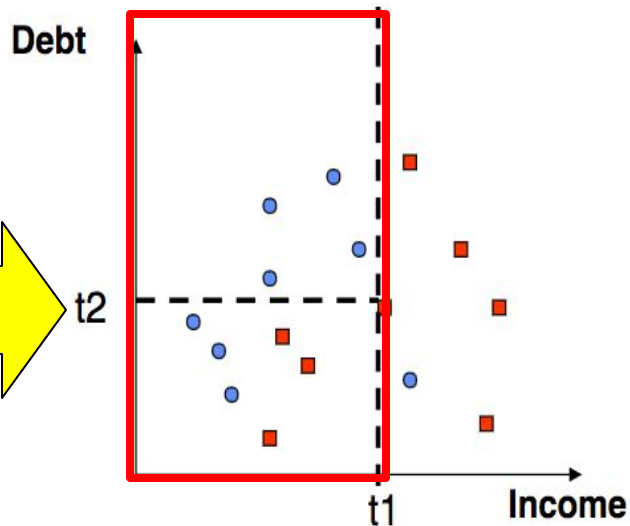
# TREE INDUCTION EXAMPLE

- Split 1



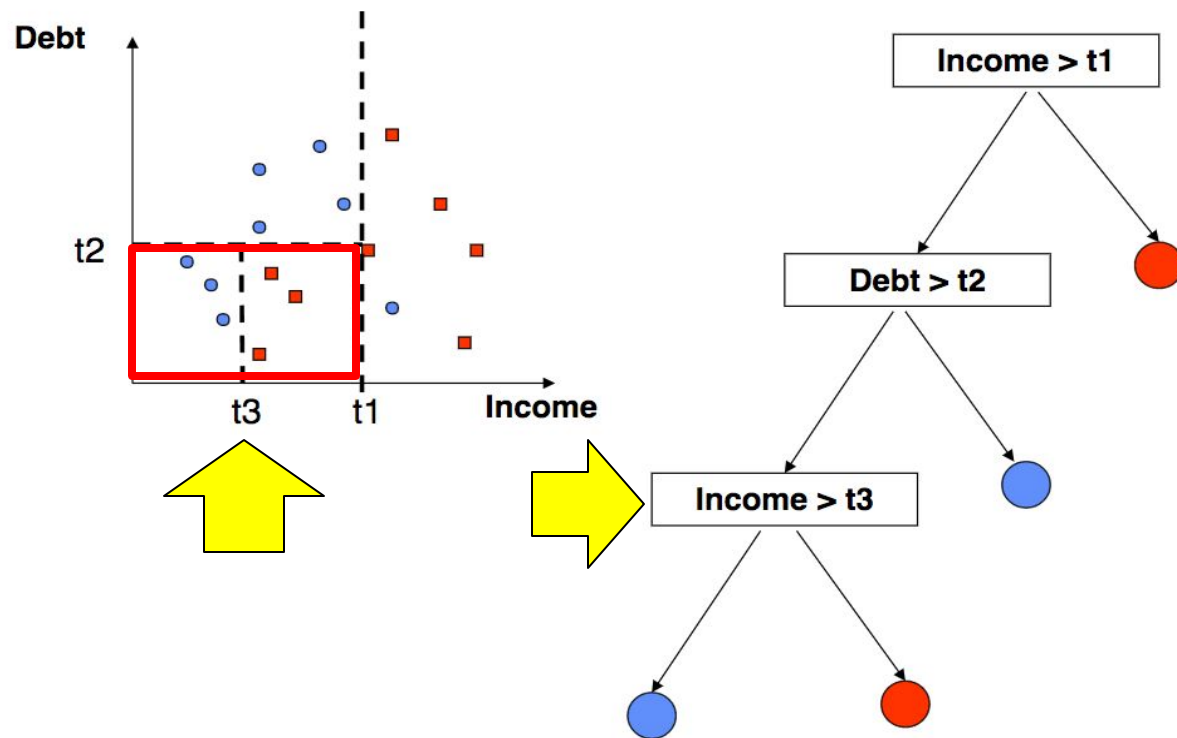
# TREE INDUCTION EXAMPLE

- Split 2



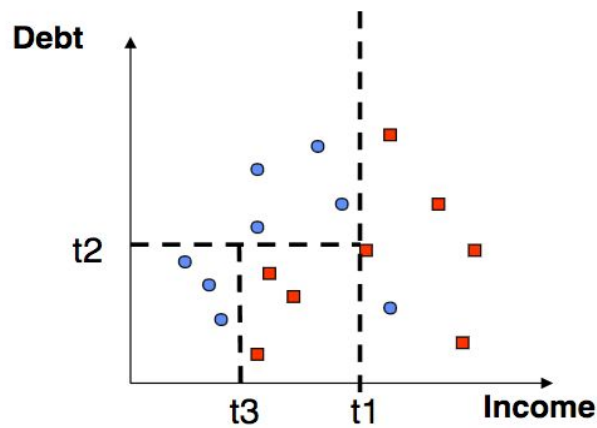
# TREE INDUCTION EXAMPLE

- Split 3

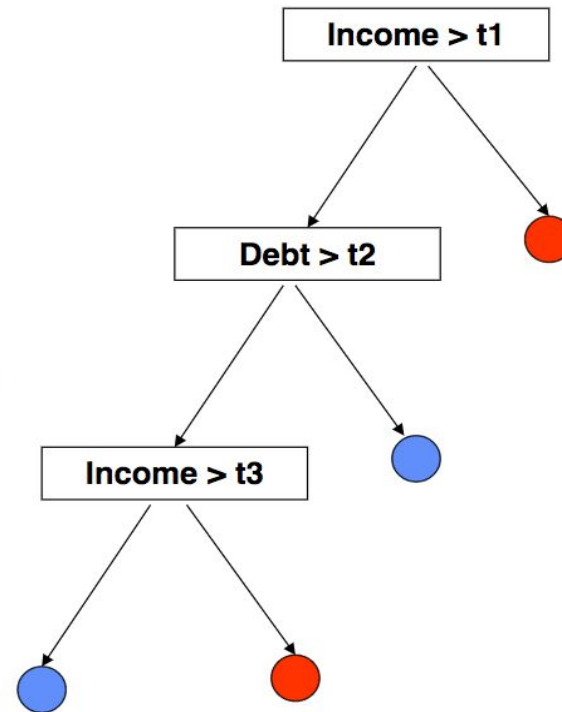


# TREE INDUCTION EXAMPLE

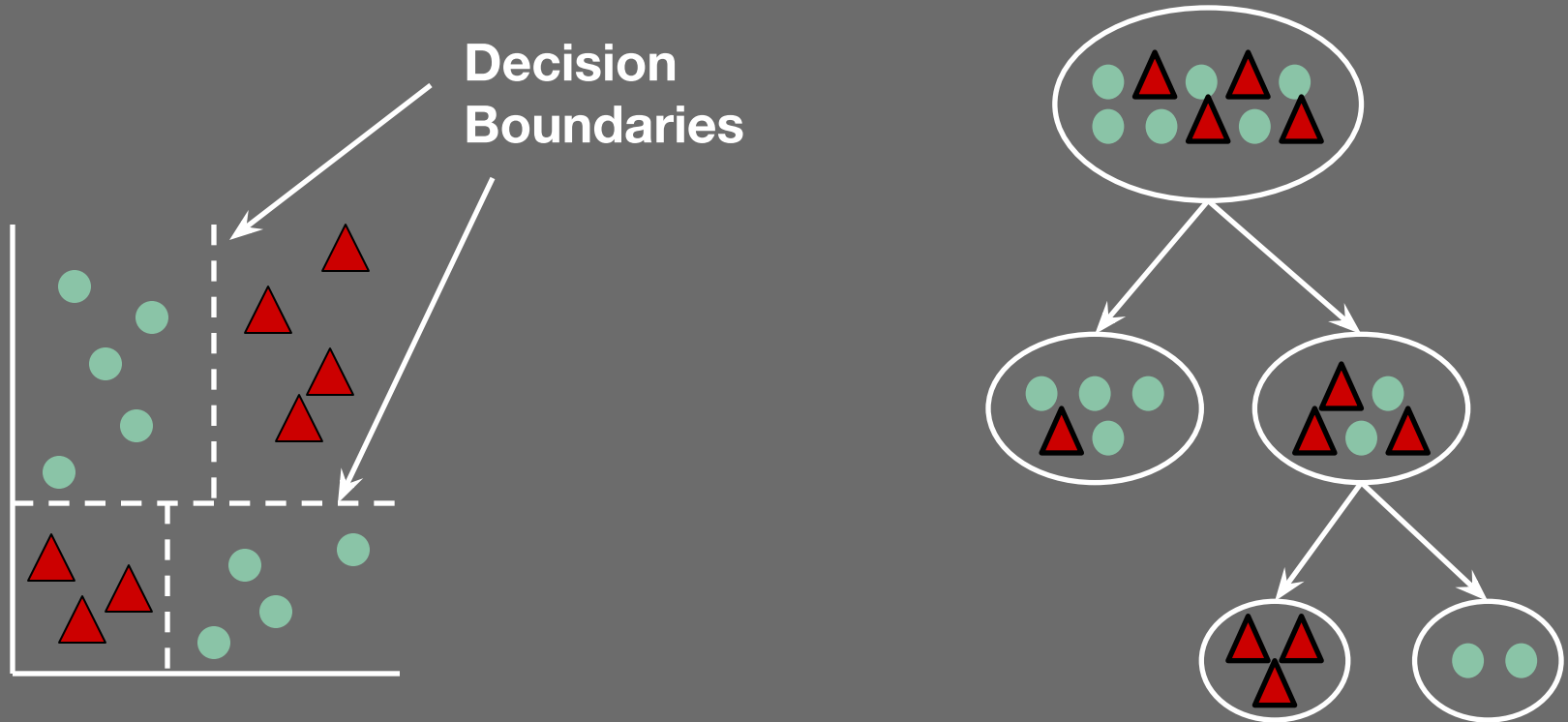
Decision Boundaries



Resulting model



# DECISION TREE MODEL



# CLASSIFICATION ALGORITHMS

- Decision Tree
- k-Nearest-Neighbor
- Naïve Bayes
- Logistic Regression
- Support Vector Machines
- Neural Networks
- Random Forest
- Many others ...



# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - Categories of Machine Learning Techniques
  - Building and Applying a Model
  - Classification
  - **Regression**
  - Cluster Analysis
- Spark MLlib
- Assignments

# REGRESSION

- Goal: Predict numeric value given input data
  - Target is numeric variable

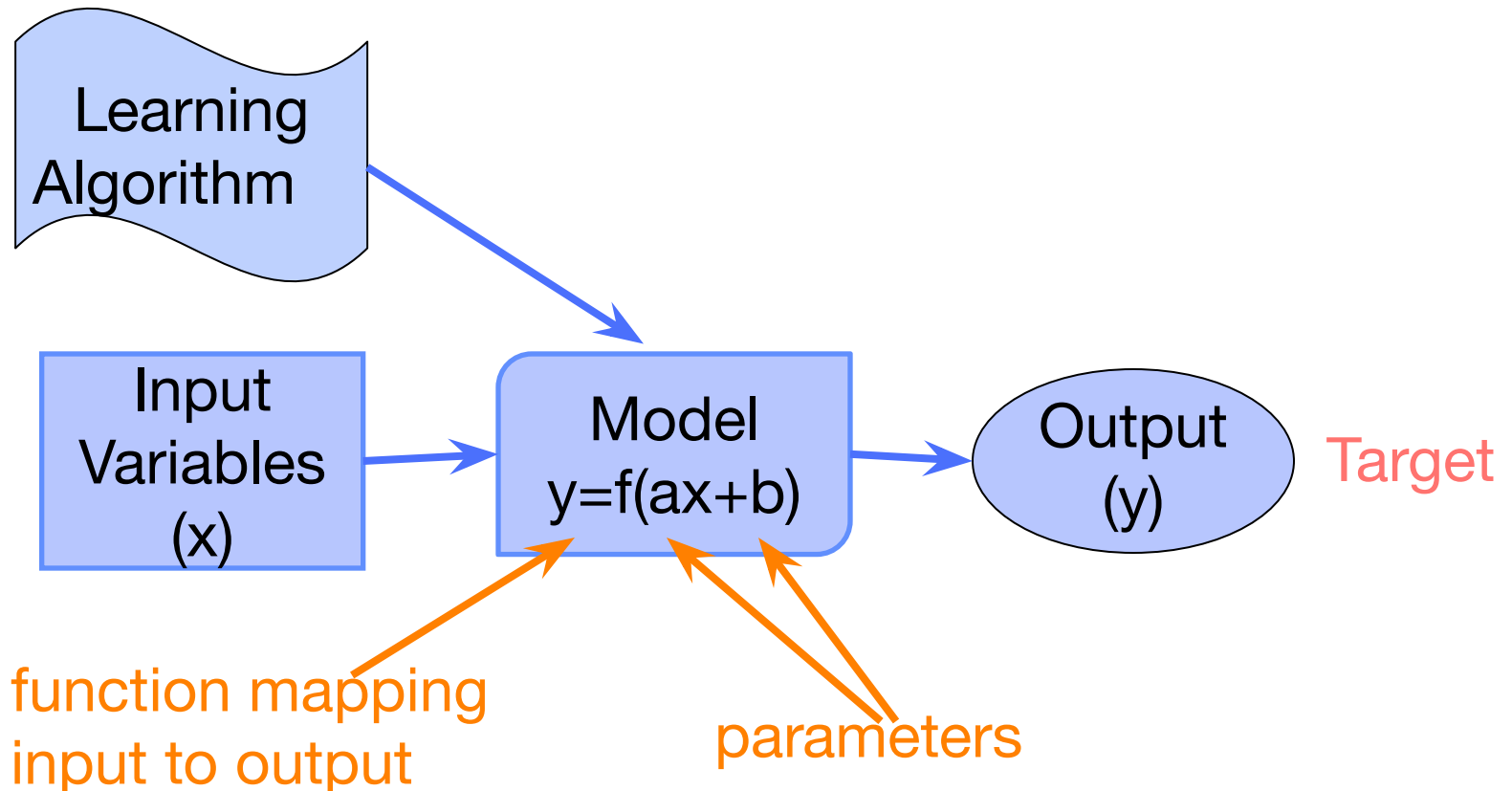


[www.wallstreetpoint.com](http://www.wallstreetpoint.com)

- Examples
  - Predict price of stock
  - Estimate demand for a product based on time of year
  - Determine risk of loan application
  - Predict amount of rain

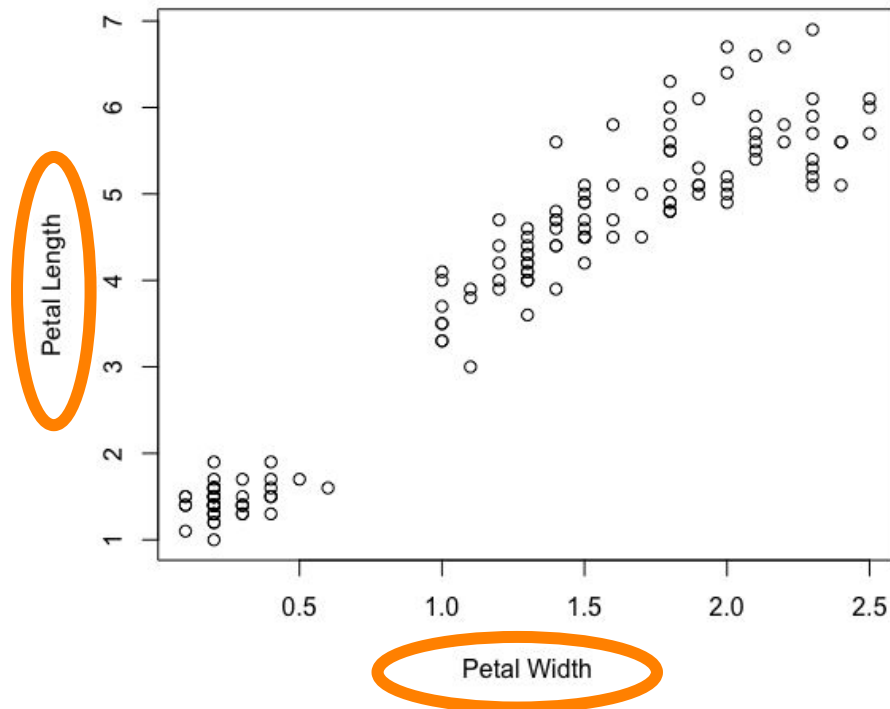
# BUILDING REGRESSION MODEL

- Goal: Get model outputs to match targets
- Training model: Adjust model parameters to minimize difference between outputs and targets



# REGRESSION

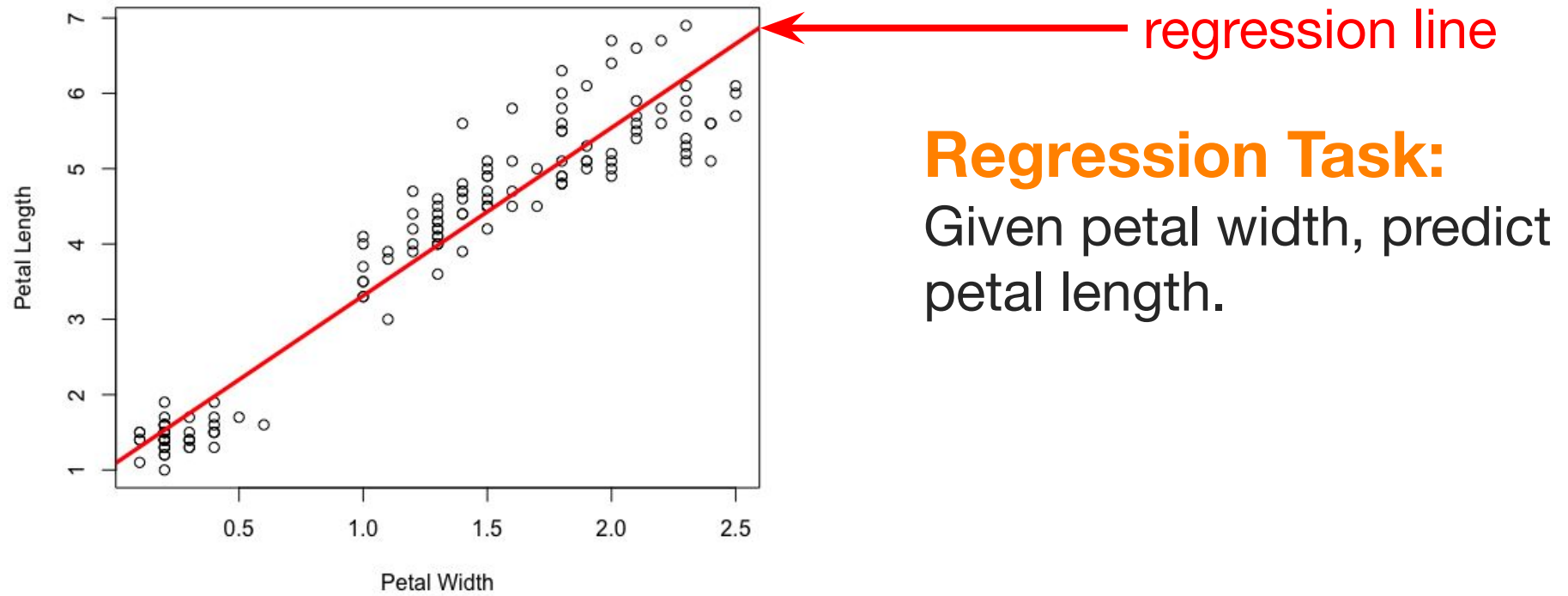
Models relationship between numerical output & input variables



**Regression Task:**  
Given petal width, predict petal length.

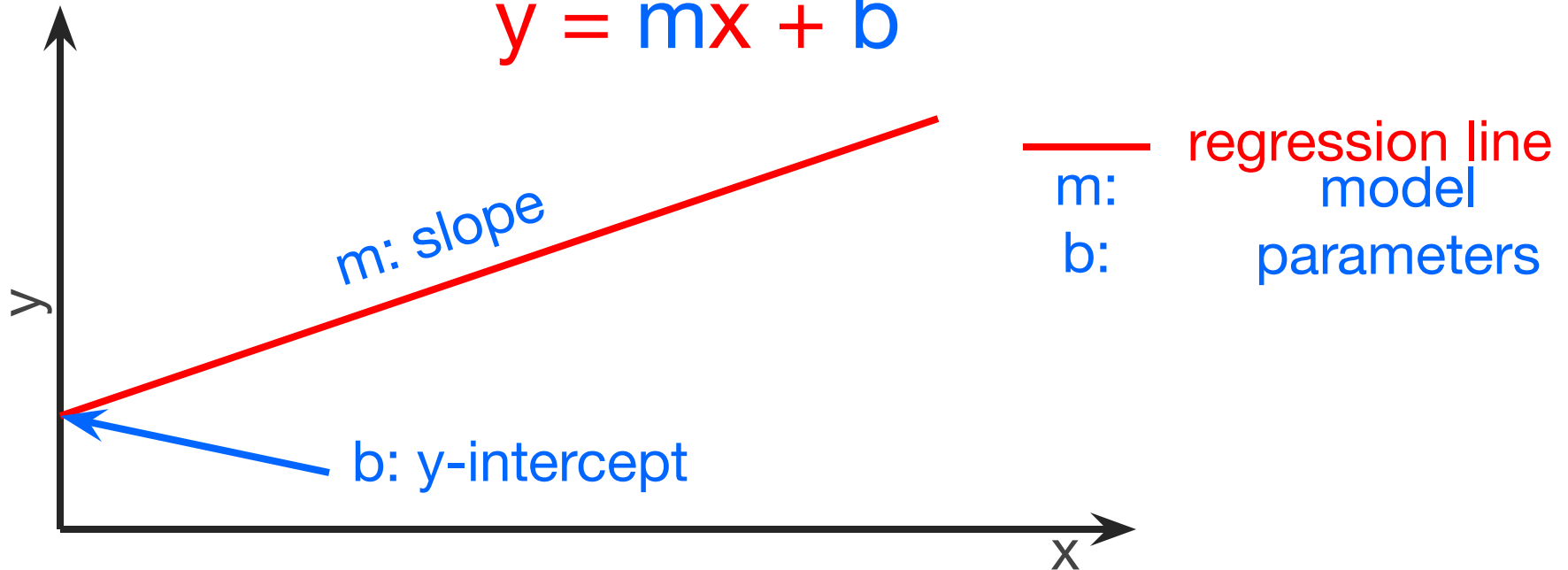
# LINEAR REGRESSION

Models relationship between numerical output & input variables as a *linear* function



# LINEAR REGRESSION

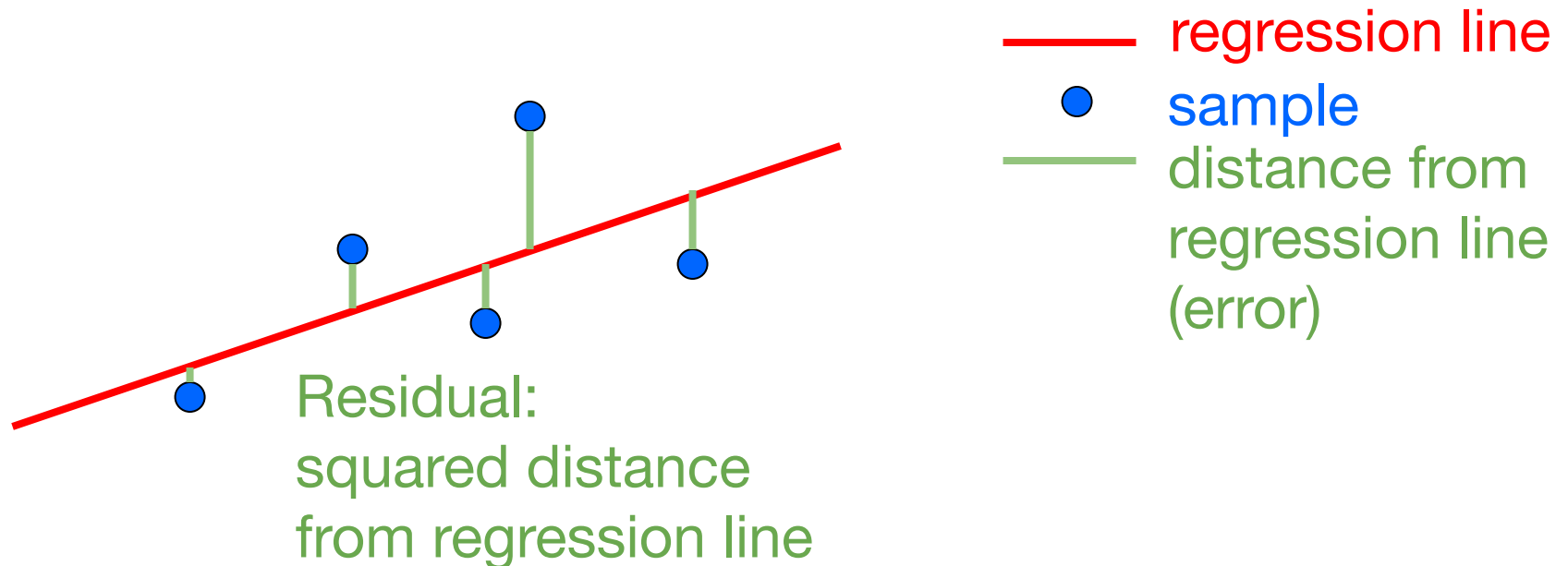
$$y = mx + b$$



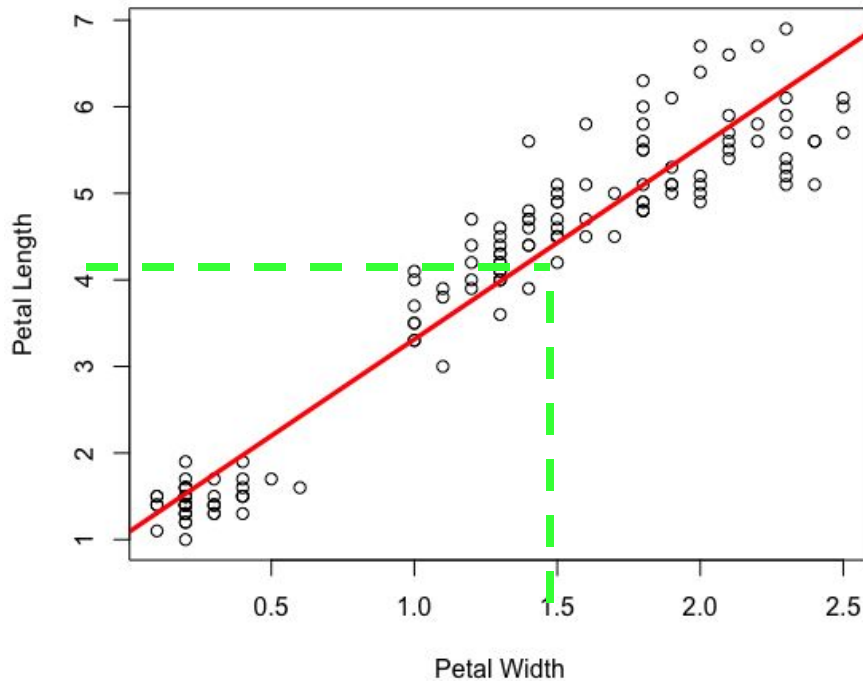
Training linear regression model adjusts model parameters to fit samples

# LINEAR REGRESSION

Goal: Find regression line that minimizes sum of residuals



# LINEAR REGRESSION MODEL



## Applying model:

Given petal width = 1.5 cm

Prediction is:

petal length = 4.5 cm

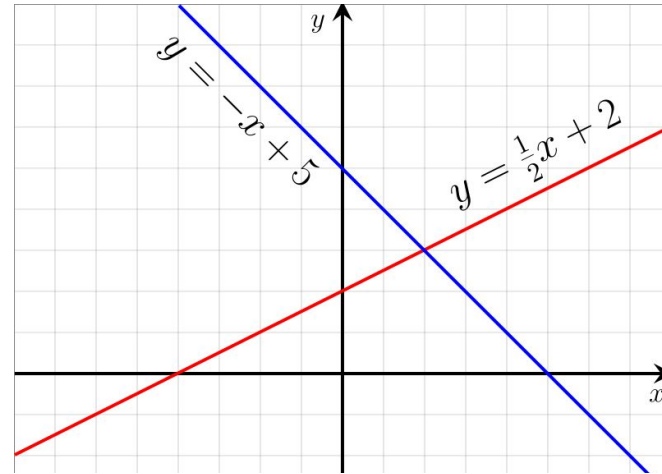


# LINEAR REGRESSION

**Special case (2D):**

$$y = mx + b$$

↑                      ↑  
slope                  y-intercept



**General case:**

$$y = w_0 + w_1x_1 + \dots + w_nx_n$$

Output

Inputs: features we are trying to model output y with

Weights: parameters we need to find to model

# REGULARIZATION

- Used to address overfitting
- Idea
  - Constrain or shrink (“regularize”) model parameters
  - To control model complexity by reducing variance of model
- Method
  - Add penalty term to error function used to train model
    - To discourage large values of parameters

# RIDGE REGRESSION

- Linear regression with ridge regularization
- Model is trained to minimize

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_2^2$$

- Regularization is L2-norm of parameter vector
- $\alpha$  is complexity parameter that controls amount of penalty
  - $\alpha$  is regularization strength
- Weights for least important parameters are shrunk

# LASSO REGRESSION

- Linear regression with lasso regularization
  - LASSO: least absolute shrinkage and selection operator
- Model is trained to minimize

$$\min_w ||Xw - y||_2^2 + \alpha ||w||_1$$

- Regularization is L1-norm of parameter vector
- $\alpha$  is complexity parameter that controls amount of penalty
- L1 penalty can force some parameters to be 0
- Lasso performs variable selection and can yield sparse models

# ELASTIC REGRESSION

- Linear regression with both L2-norm and L1-norm regularization
  - Lasso sets weights of some parameters to 0
    - Dependent on data and can be unstable
  - So combine lasso and ridge

# REGRESSION USING DECISION TREE

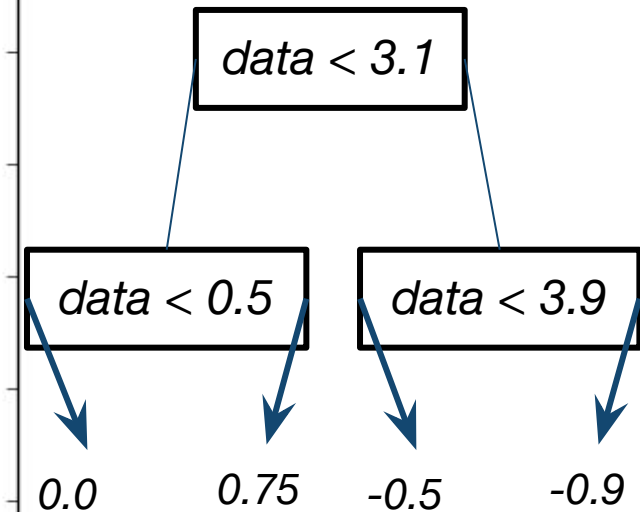
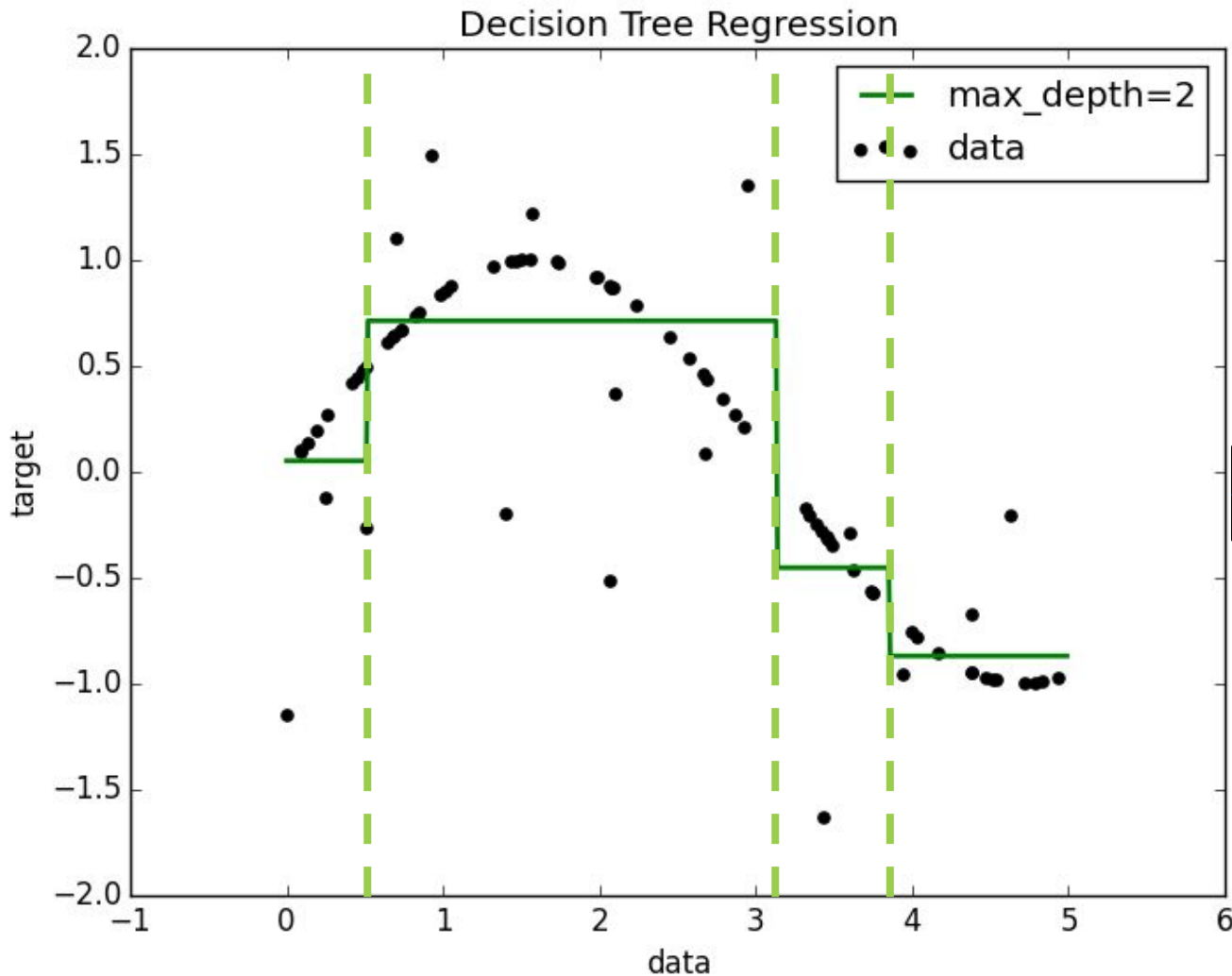
- Classification

- Feature space is split into regions
- Category of majority of samples at leaf node is predicted label for new sample

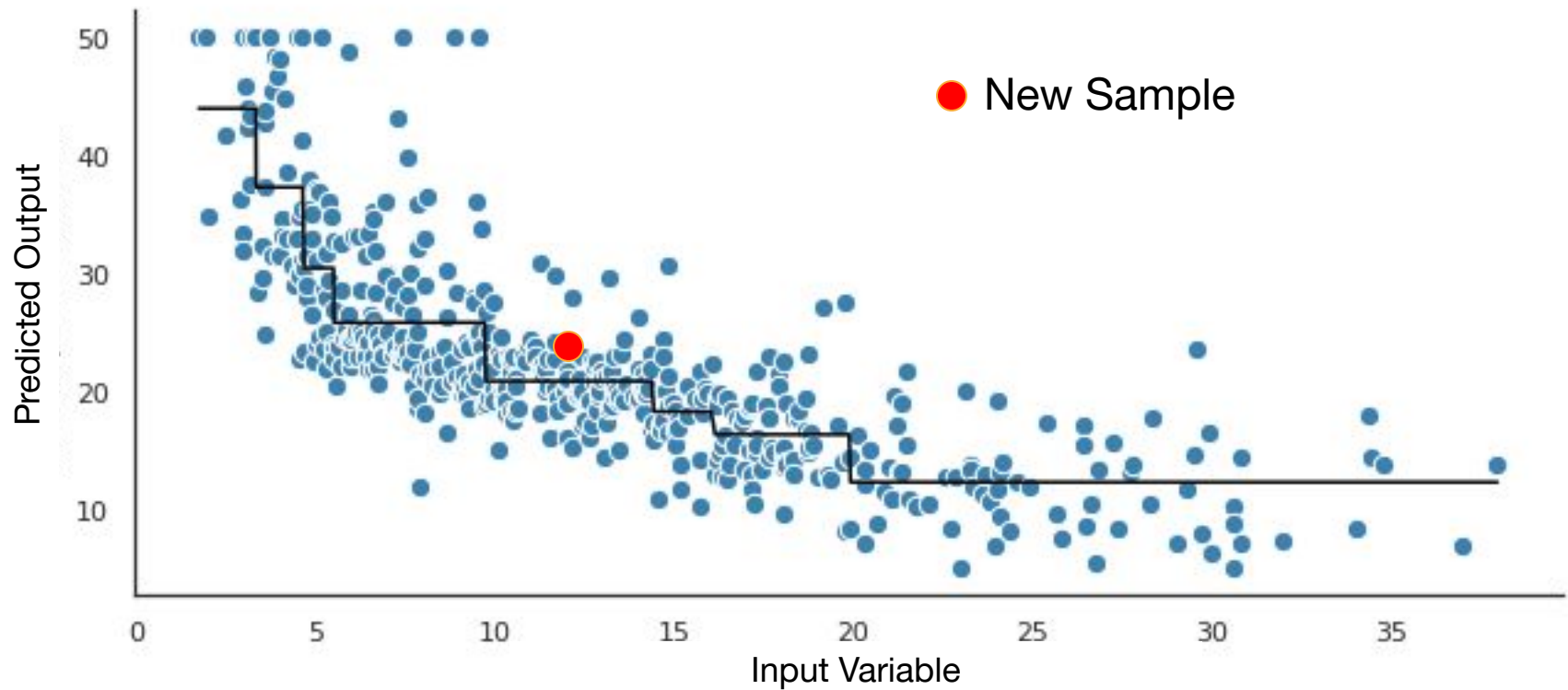
- Regression

- Feature space is split into discrete regions
- Average of samples at leaf node is predicted value for new sample

# REGRESSION USING DECISION TREE



# REGRESSION USING DECISION TREE





# REGRESSION USING DECISION TREE

- How to determine best split at each node?
- Classification
  - Want to minimize impurity of node (i.e., want split that results in most homogeneous subsets)
  - Use impurity measure (e.g., Gini index)
- Regression
  - Want to minimize variance of target values
  - Use variance reduction measure (e.g., mean squared error – MSE)

# REGRESSION ALGORITHMS

- Linear Regression
  - Ridge Regression
  - Lasso Regression
  - ElasticNet
- Decision Tree Regression
- k-Nearest Neighbor Regression
- Neural Networks
- Many others ...

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- **Modeling**
  - Categories of Machine Learning Techniques
  - Building and Applying a Model
  - Classification
  - Regression
  - **Cluster Analysis**
- Spark MLlib
- Assignments

# CLUSTER ANALYSIS

- Goal: Organize similar items into groups

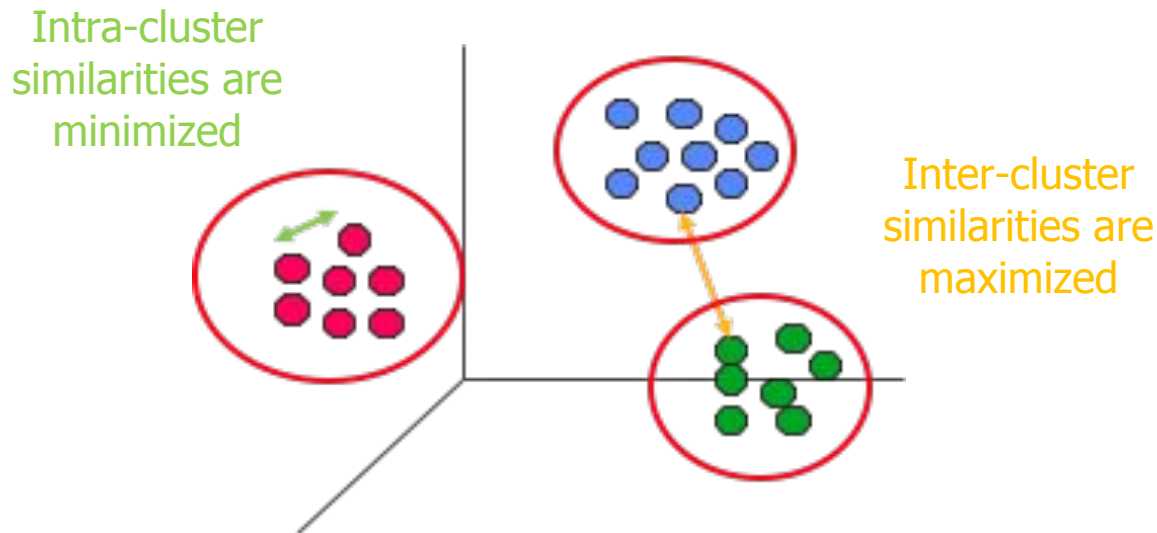


<http://www.bostonlogic.com/blog/2014/01/segment-your-leads-to-get-better-results/>

- Examples
  - Group customer base into segments for effective targeted marketing
  - Identify areas of similar topography (desert, grass, etc.)
  - Categorize different types of tissues from medical images
  - Discover crime hot spots

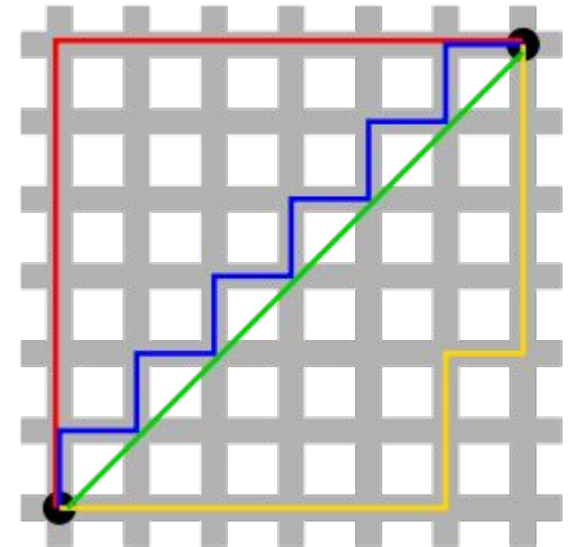
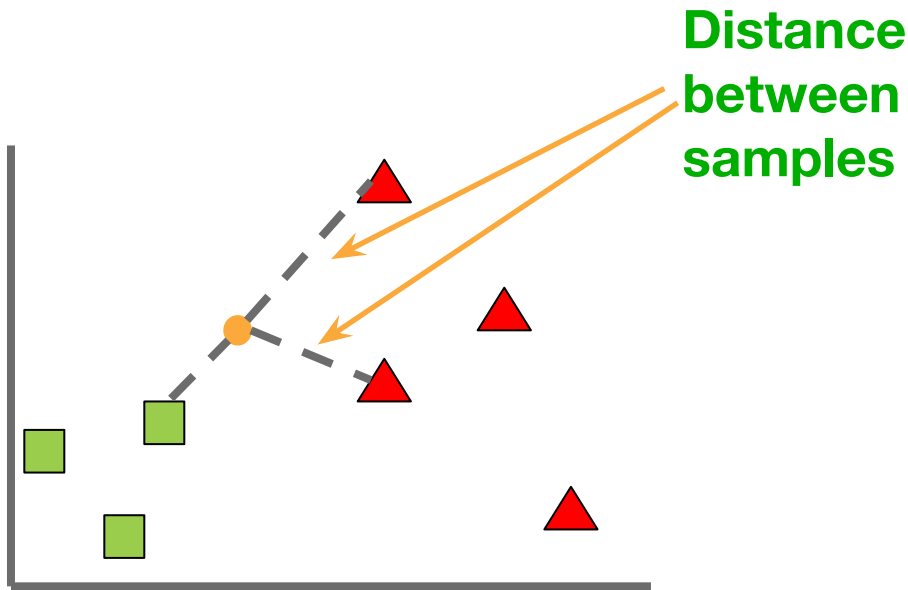
# CLUSTER ANALYSIS

- Cluster analysis divides data into groups
  - Grouping is based on some similarity measure.
  - Samples within a cluster are more similar to each other than to samples in other clusters.



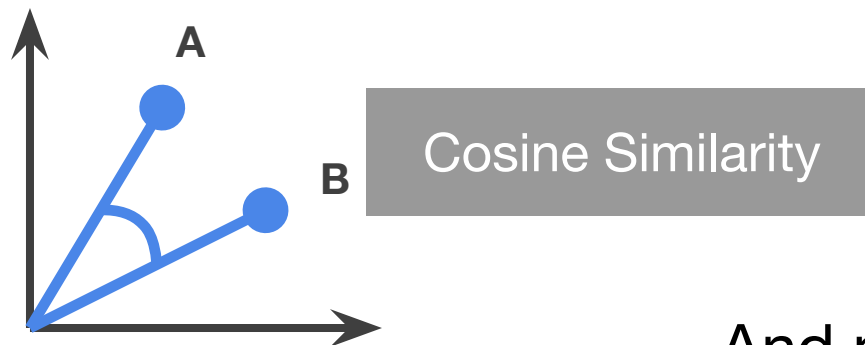
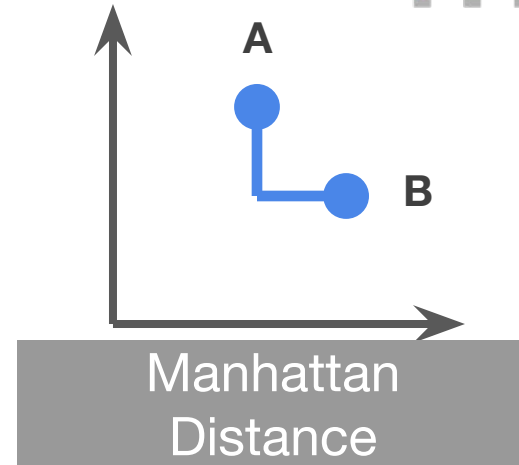
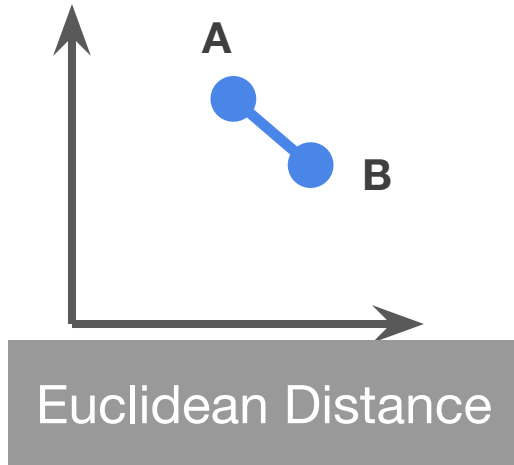
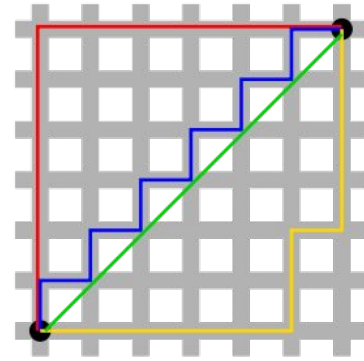
<http://www-users.cs.umn.edu/~kumar/dmbook/index.php>

# DISTANCE MEASURE



- Distance measures used to measure similarity
  - Euclidean (green line)
  - Manhattan (red, blue, yellow lines)
  - Many, many more!

# SIMILARITY MEASURES



And many others...

# CLUSTER ANALYSIS

- Cluster analysis is *unsupervised*
  - Samples are unlabeled: Target is unknown or unavailable
- In general, there is no 'correct' clustering
  - 'Best' set of clusters is dependent on how clusters will be used
- Clusters don't come with labels
  - User must label clusters by analyzing samples in each cluster
- Thus, interpretation and analysis are required to make sense of clustering results!

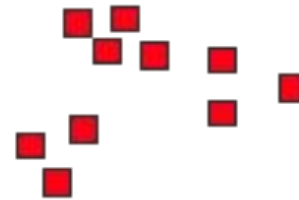


# CLUSTER ANALYSIS

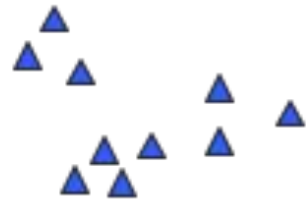
- Interpretation is required to analyze results:



*How many clusters?*



*Two Clusters*



*Four Clusters*



*Six Clusters*



<http://www-users.cs.umn.edu/~kumar/dmbook/index.php>

# $k$ -MEANS CLUSTERING

- Partitional
  - Clusters are divided into non-overlapping subsets
- Centroid-Based
  - Cluster represented by central vector
- Simple, classic clustering technique
  - Data points are grouped into  $k$  clusters
  - Cluster defined by cluster mean

- Algorithm:

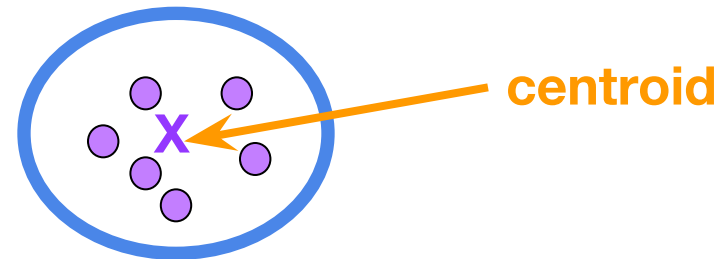
Select  $k$  initial *centroids* (cluster centers)

Repeat

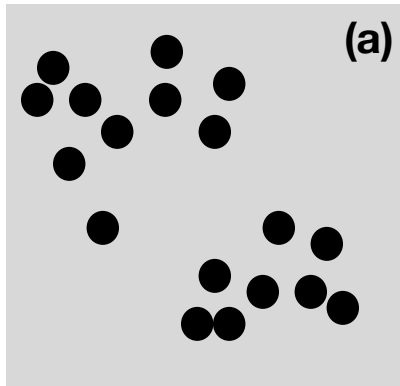
    Assign each sample to closest centroid

    Calculate mean of cluster to determine new centroid

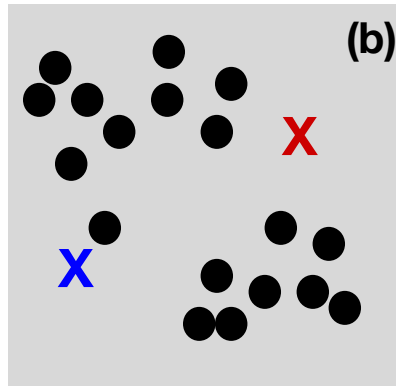
Until some stopping criterion is reached



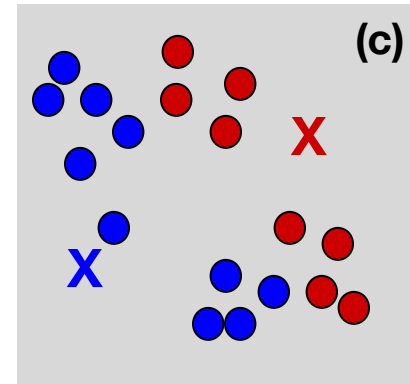
# $k$ -MEANS CLUSTERING ILLUSTRATION



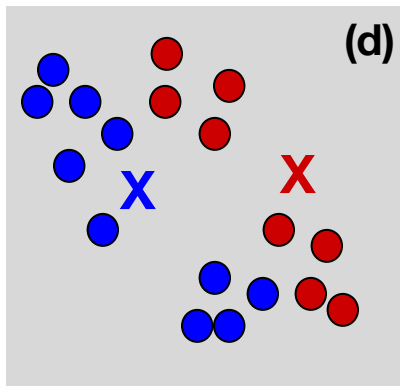
Original samples



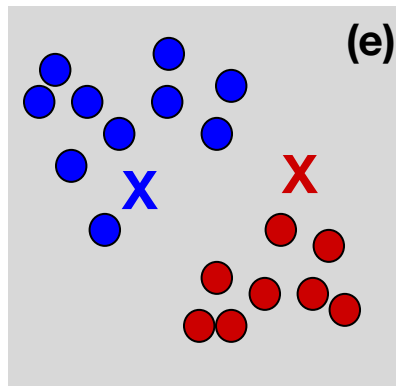
Initial Centroids



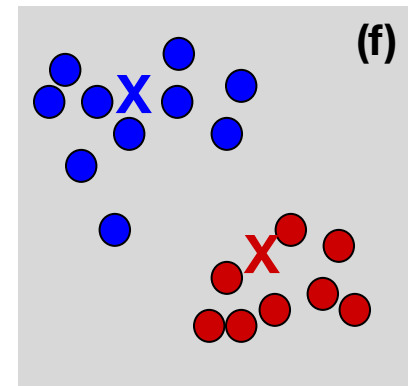
Assign Samples



Re-calculate Centroids



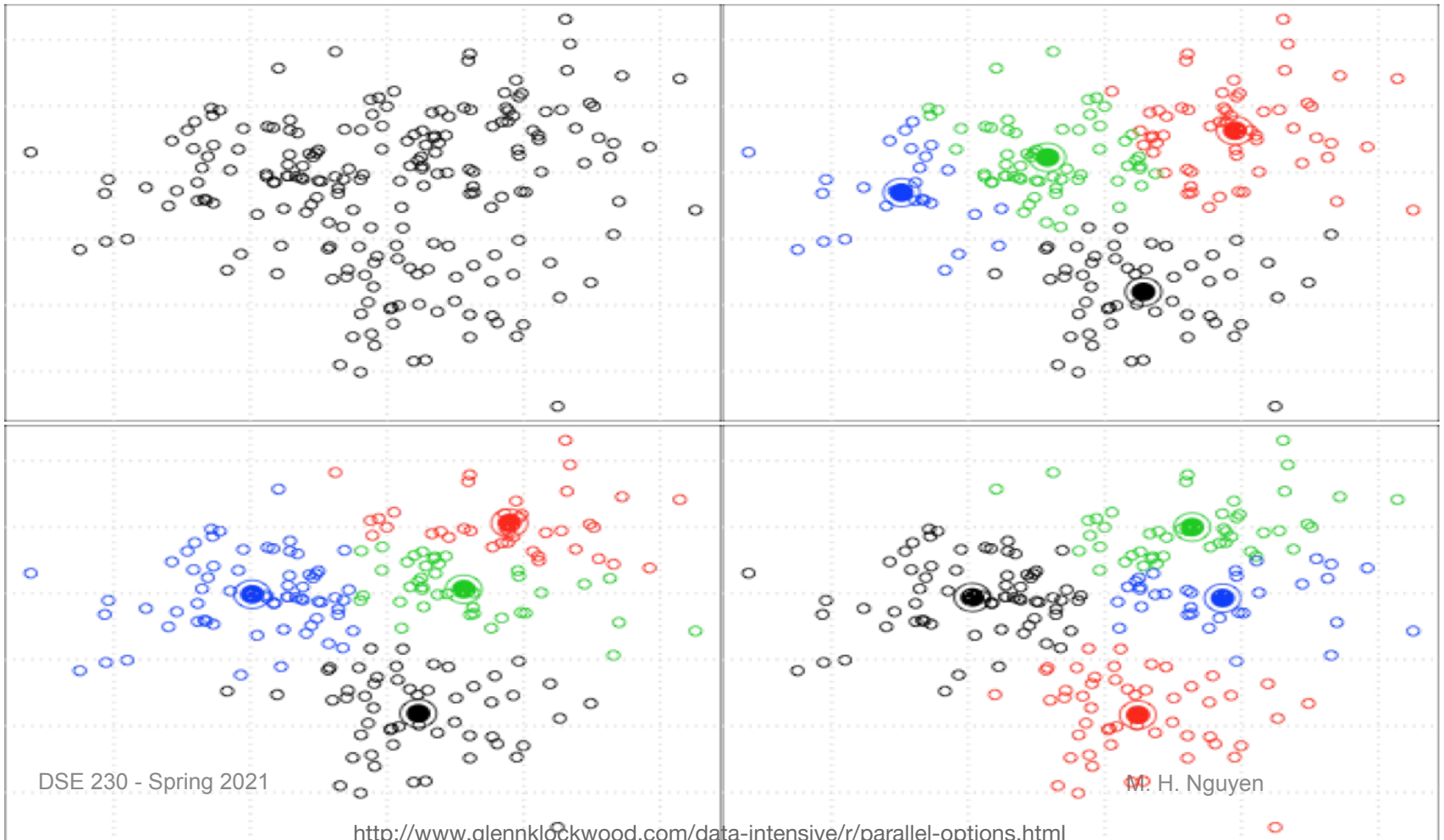
Assign Samples



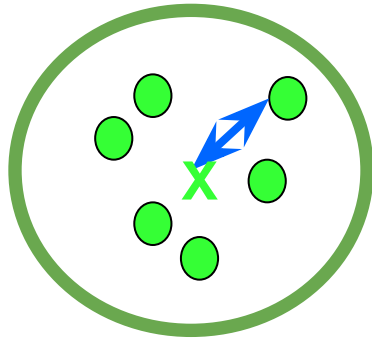
Re-calculate Centroids

# $k$ -MEANS CLUSTERING DETAILS

- The number of clusters,  $k$ , must be specified
- ‘Closeness’ between two samples determined using similarity measure (Euclidean distance, cosine similarity, etc.)
- Clustering results are sensitive to initial centroids



# EVALUATING CLUSTERING RESULTS



error = distance between sample & centroid  
squared error =  $\text{error}^2$

Sum of squared errors  
between all samples &  
centroid

Sum over all clusters



WSSE = Within-Cluster Sum  
of Squared Error

# EVALUATING CLUSTERING RESULTS

- Within-Cluster Sum of Squared Error (WSSE)
- For each sample, error is distance to centroid. Then, WSSE is computed as:

$$WSSE = \sum_{i=1}^K \sum_{x \in C_i} \|x - m_i\|^2$$

$x$ : data sample in cluster  $C_i$

$m_i$ : cluster centroid (i.e., mean of cluster)

$\|x - m_i\|^2$ : Euclidean distance between  $m_i$  and  $x$

# CHOOSING INITIAL CENTROIDS

- Problem:
  - Final clusters are sensitive to initial centroids
- Approaches:
  - Perform multiple runs with different initial centroids, and choose best results
  - Randomly select more than  $k$  initial centroids. Then select among those the most widely separated
  - Apply hierarchical clustering. Centroids of  $k$  clusters are used as initial centroids for  $k$ -means

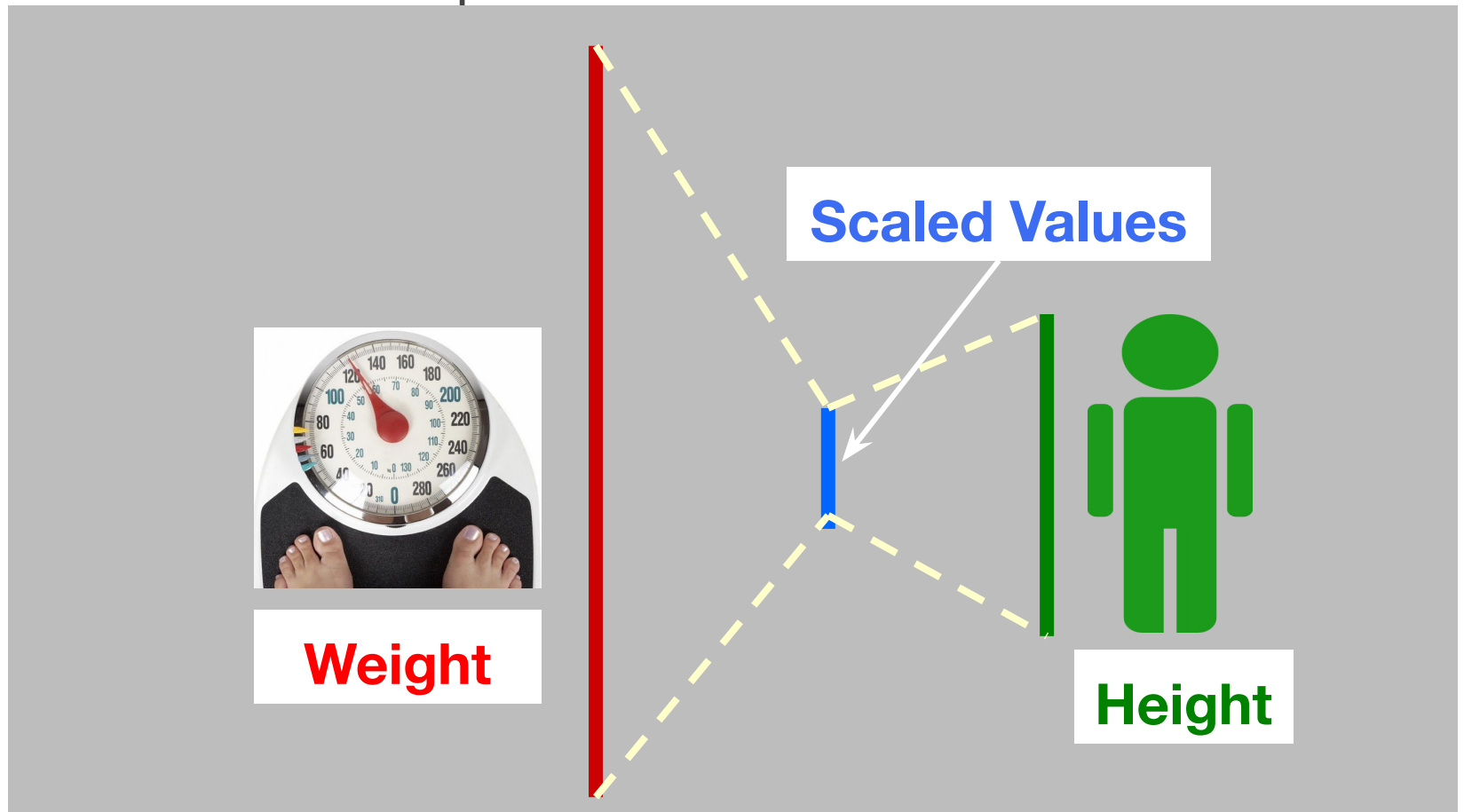
# $k$ -MEANS++ and $k$ -MEANS ||

- Choose initial centroids more selectively
- Idea:
  - Choose initial centroids that are far apart from one another
- Algorithm
  - Choose first centroid at random
  - For each sample  $\mathbf{x}$ , compute  $D(\mathbf{x}) = \text{dist}(\mathbf{x}, \text{closest\_centroid})$
  - Choose new centroid  $\mathbf{y}$ 
    - $\mathbf{y}$  is chosen with probability proportional to  $D(\mathbf{y})^2$
  - After  $k$  centroids are selected, continue with standard  $k$ -means
- $k$ -means|| is parallel implementation of  $k$ -means++



# SCALING INPUT VALUES

Scale to prevent distortion of results

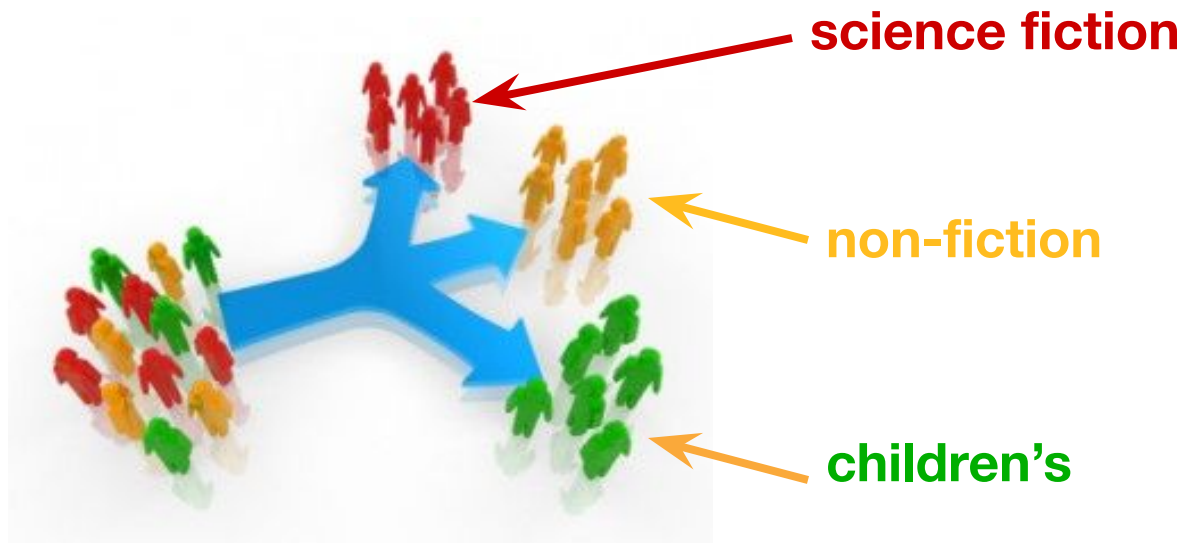


# VARIATIONS OF $k$ -MEANS

- $k$ -medians
  - Uses median instead of mean as cluster centroid
- $k$ -medoids
  - Uses mediod instead of mean as cluster centroid
    - Mediod: actual sample point
  - Minimizes sum of dissimilarities between samples and cluster centroids

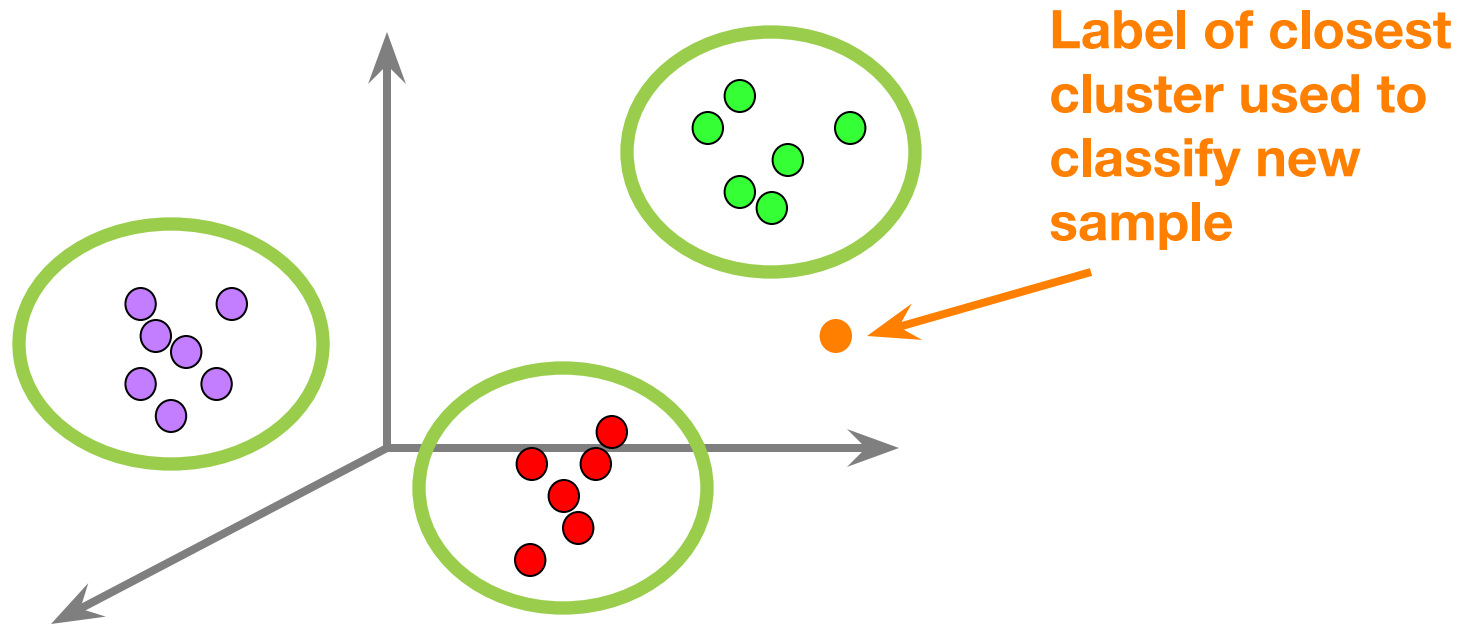
# USES OF CLUSTER RESULTS

- Data segmentation
  - Analysis of each segment can provide insights



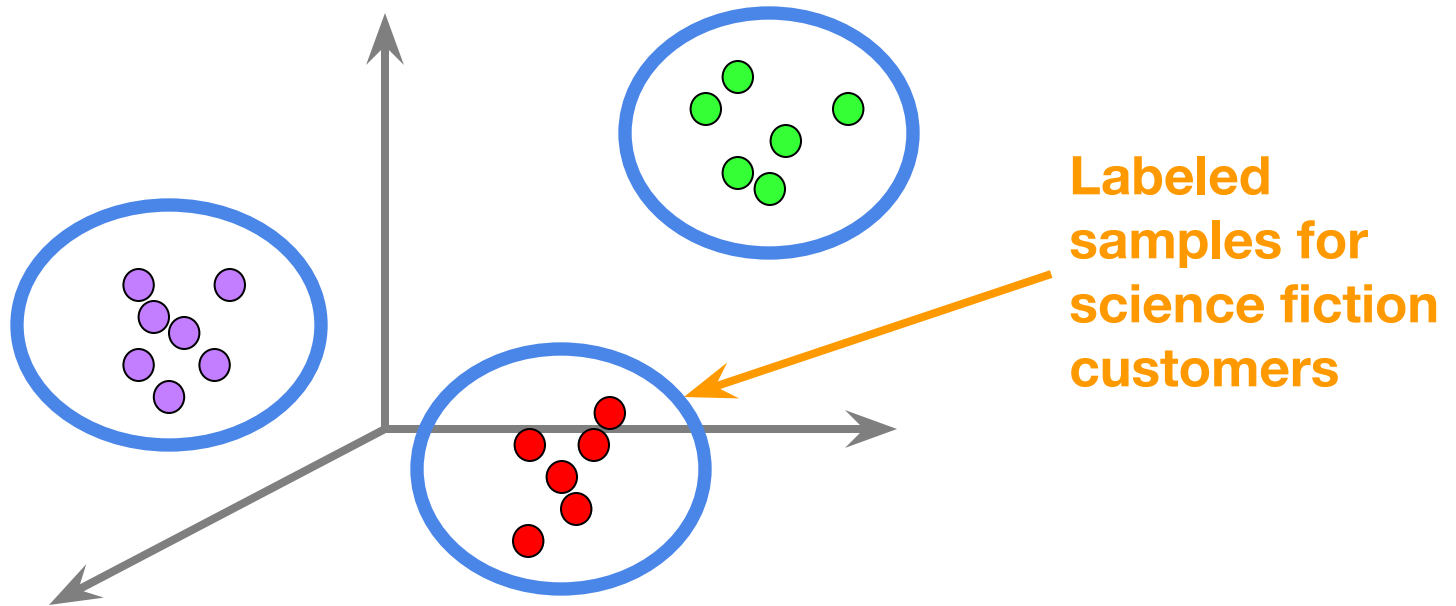
# USES OF CLUSTER RESULTS

- Categories for classifying new data
  - New sample assigned to closest cluster



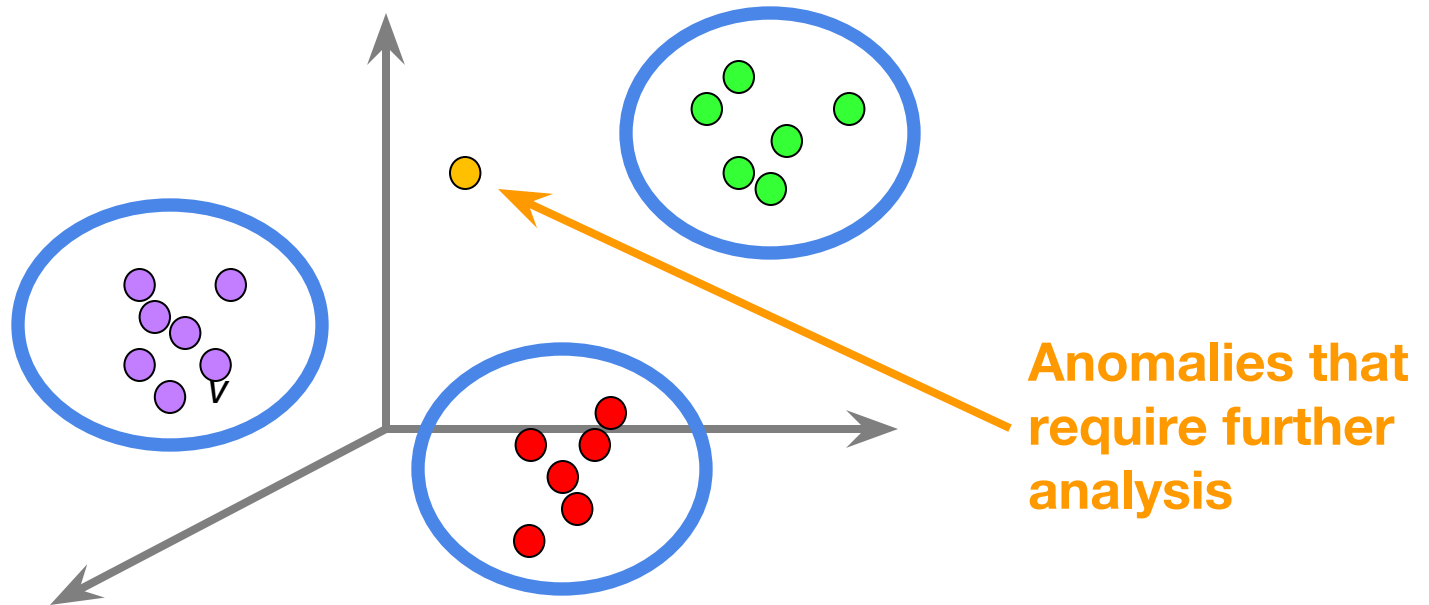
# USES OF CLUSTER RESULTS

- Labeled data for classification
  - Cluster samples used as labeled data



# USES OF CLUSTER RESULTS

- Basis for anomaly detection
  - Cluster outliers are anomalies



# TYPES OF CLUSTERING ALGORITHMS

- Exclusive vs. Overlapping vs. Probabilistic
  - **Exclusive:** A sample belongs only to one cluster.
  - **Overlapping:** A sample can fall into several clusters
  - **Probabilistic:** A sample belongs to a cluster with a certain probability.
    - Also referred to as “soft clustering” or “fuzzy clustering”
- Partitional vs. Hierarchical
  - **Partitional:** Clusters are divided into non-overlapping subsets
  - **Hierarchical:** Clusters are nested and organized in a hierarchical tree.

# CLUSTER ANALYSIS ALGORITHMS

- $k$ -Means
- $k$ -Medians
- $k$ -Medoids
- Hierarchical clustering
- Density-based clustering
- Gaussian mixture models
- Others ...

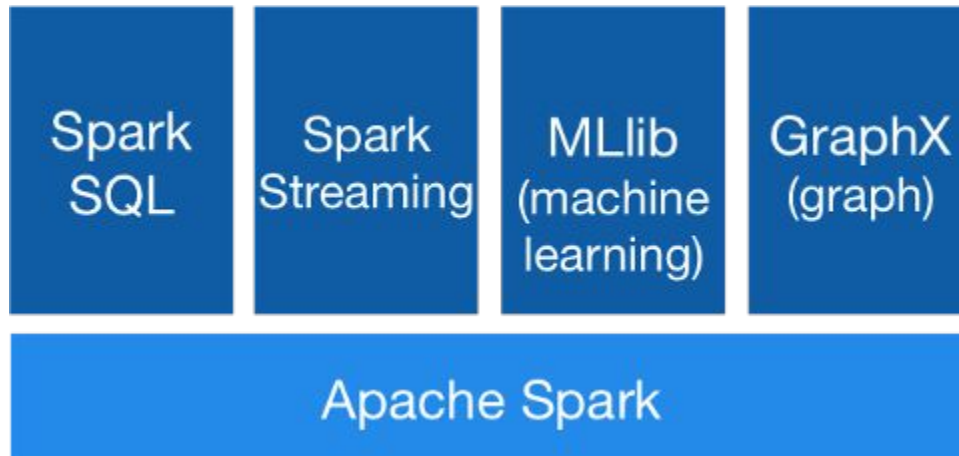


# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- Modeling
  - Categories of Machine Learning Techniques
  - Building and Applying a Model
  - Classification
  - Regression
  - Cluster Analysis
- **Spark MLlib**
- Assignment

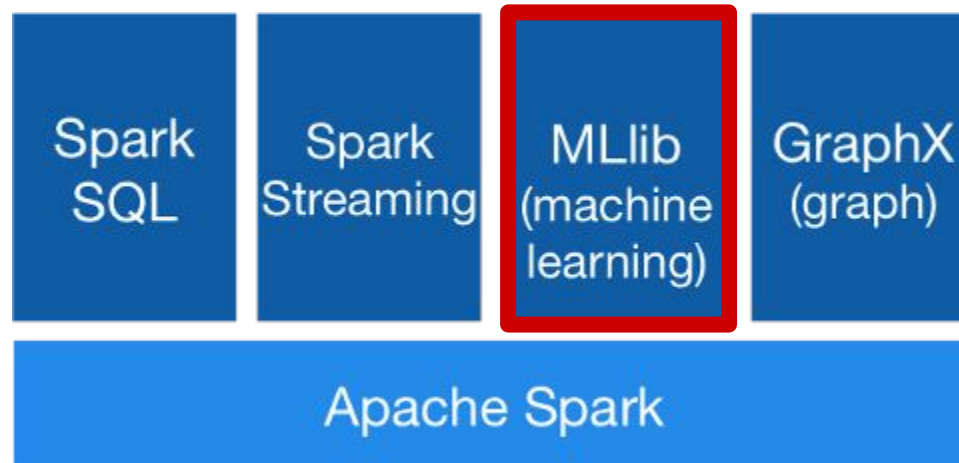
# SPARK AS UNIFIED PLATFORM

- Goals: speed, ease of use, generality, **unified platform**



- Provides unified platform for various analytics processing
- **Spark engine** provides core capabilities for distributed processing
- **Spark libraries** provide additional higher-level functionality for diverse workloads

# SPARK MLLIB



- Machine Learning
  - Scalable machine learning library
  - Distributed implementations of machine learning algorithms and utilities
  - Has APIs for Scala, Java, Python, and R

# SPARK MLLIB ALGORITHMS

- Machine Learning
  - Classification, regression, clustering, etc.
  - Evaluation metrics
- Statistics
  - Summary statistics, sampling, etc.
- Utilities
  - Dimensionality reduction, transformation, etc.
- ML Pipelines
  - Similar to scikit-learn

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- Modeling
- Spark MLlib
- **Assignments**

# SESSION 2 ASSIGNMENTS

- Reading Assignment
  - A Data Scientist's Guide to Apache Spark
- Project Proposal
  - Present in class during Session 3 (2021-05-01)
  - Discuss
    - Problem to address
    - Dataset
    - Analysis task planned
    - Insights you hope to gain
    - Potential challenges
- Project Teams
  - Sign up sheet: Link posted on Piazza
  - Sign up by Monday 2021-04-19 at 11:59pm Pacific Time

# SESSION 2 ASSIGNMENTS

- Programming Assignment
  - WordCount on Amazon Reviews
    - ▢ Data file: BookReviews\_1M.txt
    - ▢ Use PySpark DataFrame (not RDD)
    - ▢ Find top 100 words based on count
    - ▢ Find mean and standard deviation of execution times over 3 runs
      - ❖ Using 1 core, 2 cores, and 4 cores
  - Submit
    - ▢ Jupyter notebook (.ipynb)
    - ▢ Python script (.py)
    - ▢ Word count results (.csv): Top 100 words
      - ❖ word, count
    - ▢ Execution times results (.csv)
      - ❖ # cores, time0, ... time2, mean, stdev
  - Due Friday 2021-04-30 at 11:59pm Pacific Time

# WORDCOUNT OUTLINE

- Read data into DataFrame
- Remove punctuations and convert to lower case
- Split data into words
- Put each word in a separate row
- Filter out words with length 0
- Group rows by word to count the number of occurrences for each word
- Sort words by count
- Notes
  - Remember to copy data file to HDFS
  - Will need to copy results file from HDFS



# COPY RESULTS FROM HDFS

- In Terminal window
- List contents in HDFS # Should see <your-result-file>.csv
  - `hadoop fs -ls /S2`
- List contents of saved folder
  - `hadoop fs -ls /S2/<your-result-file>.csv/`
- Copy results from HDFS to local file system
  - `hadoop fs -copyToLocal /S2/<your-result-file>.csv/part-*.csv results_1M.csv`
- Create text file with first 101 rows from results
  - `head -n 101 <results-file> > top-results.txt`

# GETTING EXECUTION TIMES

- In notebook, execution time is printed out in cell before Spark session is stopped (next to last cell)
- Need to restart the kernel and run all cells without stopping to get accurate execution time:
  - Run -> Restart Kernel and Run All Cells
- Find mean and standard deviation of execution times over 3 runs for
  - 1 core, 2 cores, and 4 cores

```
import pyspark
from pyspark.sql import SparkSession
```

```
conf = pyspark.SparkConf().setAll([
    ('spark.master', 'local[2]'),
    ('spark.app.name', 'PySpark WordCount')])
spark = SparkSession.builder.config(conf=conf).getOrCreate()
```

Specify number of cores.  
“\*” uses all available cores



# SPARK RESOURCES

- PySpark SQL Basics Cheat Sheet
  - PDF on Canvas
- Spark Main Page
  - <https://spark.apache.org/>
- Spark Overview
  - <https://spark.apache.org/docs/latest/index.html>
- Spark Examples
  - <https://spark.apache.org/examples.html>
- Spark SQL, DataFrames and DataSets Programming Guide
  - <https://spark.apache.org/docs/latest/sql-programming-guide.html>
- Spark MLlib Programming Guide
  - <https://spark.apache.org/docs/latest/ml-guide.html>
- PySpark API Documentation
  - <https://spark.apache.org/docs/latest/api/python/index.html>
- Note: Spark version 3.1.1, Python, DataFrame API

# BIG DATA ANALYTICS

- Machine Learning Overview
- Data Exploration
- Data Preparation
- Modeling
- Spark MLlib
- Assignments