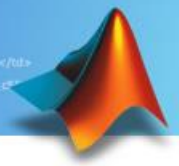
TeraSoft 鈦思科技 The Power of Driver for the Best!!

Big Data and Machine Learning Workshop

Application Engineer
Jeffrey Liu



Outline

- **Big Data Capability in MATLAB**
  - Big data in industry
  - New Big Data Capabilities in MATLAB
  - Access Big Data
  - Datastore & MapReduce
  - MATLAB on Hadoop

- **Machine Learning**
  - Overview of Machine Learning
  - Clustering and Classification
  - Applied Machine Learning in data
  - Classification Learner

Big Data in Industry

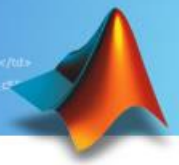
ENERGY
Asset Optimization

FINANCE
Market Risk, Regulatory

AUTO
Fleet Data Analysis

AERO
Maintenance, reliability

Medical Devices
Patient Outcomes



Challenges of Big Data

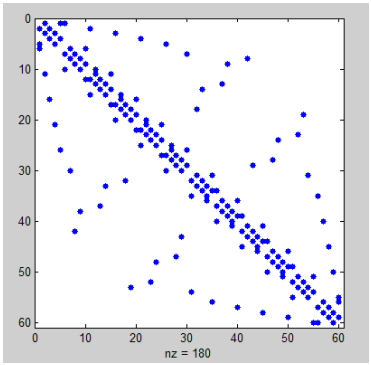
"Any collection of data sets so large and complex that it becomes difficult to process using … traditional data processing applications."
(Wikipedia)

- How to get started
- Rapid data exploration
- Development of scalable algorithms
- Use of algorithms within business systems

MATLAB and Memory

Best Practices for Memory Usage

- Use the appropriate data storage
  - Use only the precision your need
  - Sparse Matrices
  - Categorical Arrays
  - Be aware of overhead of cells and structures



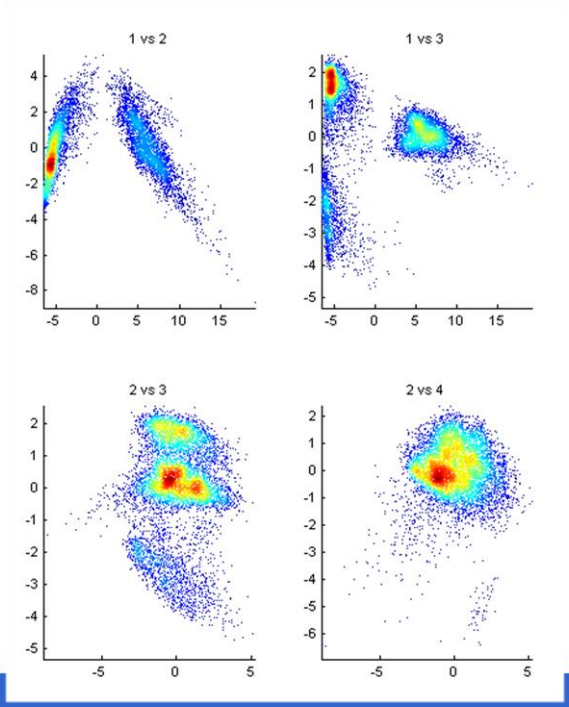
- Minimize Data Copies
  - In place operations, if possible
  - Use nested functions
  - If using objects, consider handle classes

```
function primaryFx
   x = 1;
   nestedFx;

   function nestedFx
      x = x + 1;
   end
end
```

New Big Data Capabilities in MATLAB

Memory and Data Access

- 64-bit processors
- Memory Mapped Variables
- Disk Variables
- Databases
- **Datastores** R2014b



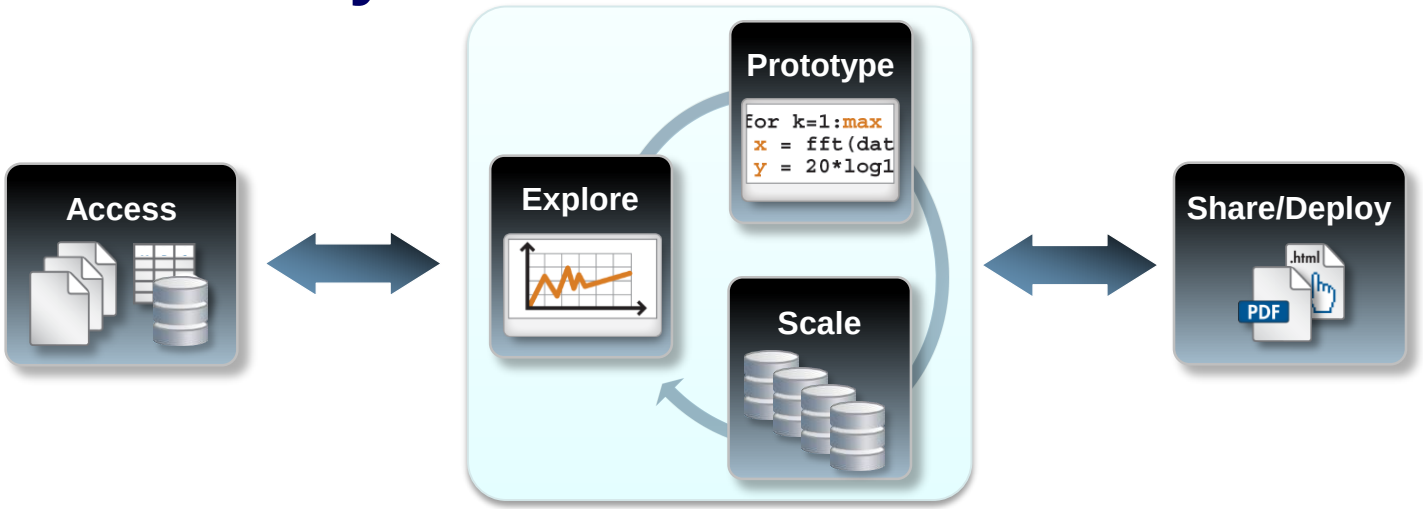
Programming Constructs

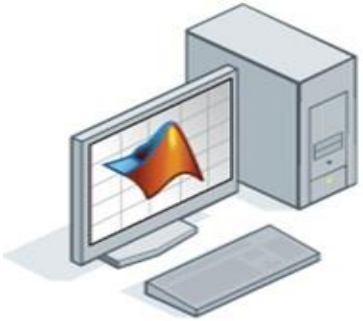
- Streaming
- Block Processing
- Parallel-for loops
- GPU Arrays
- SPMD and Distributed Arrays
- **MapReduce** R2014b

Platforms

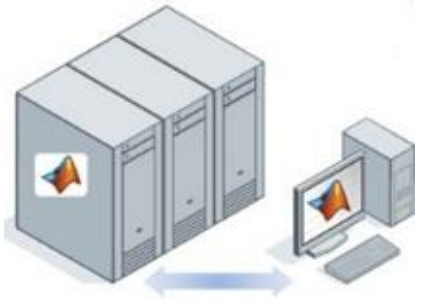
- Desktop (Multicore, GPU)
- Clusters
- Cloud Computing (MDCS on EC2)
- **Hadoop** R2014b

Big Data Analysis with MATLAB



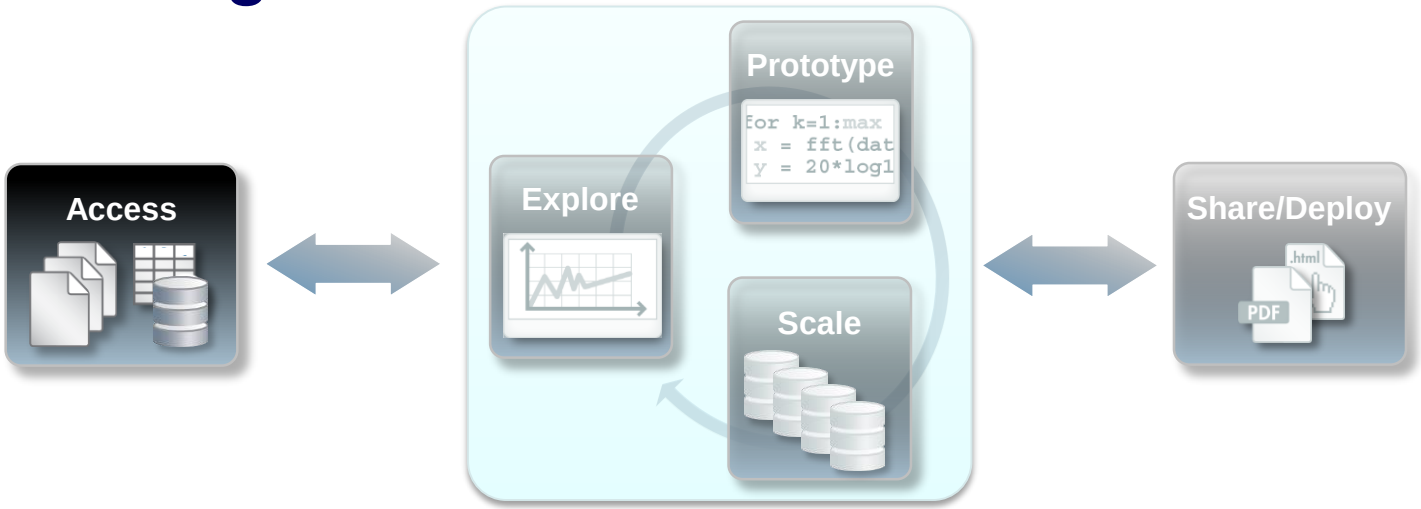


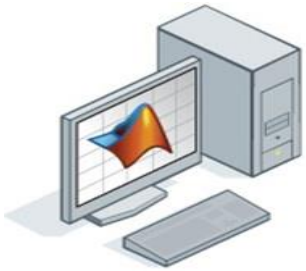
Easily get started on the desktop



Scale capacity as needed

Access Big Data





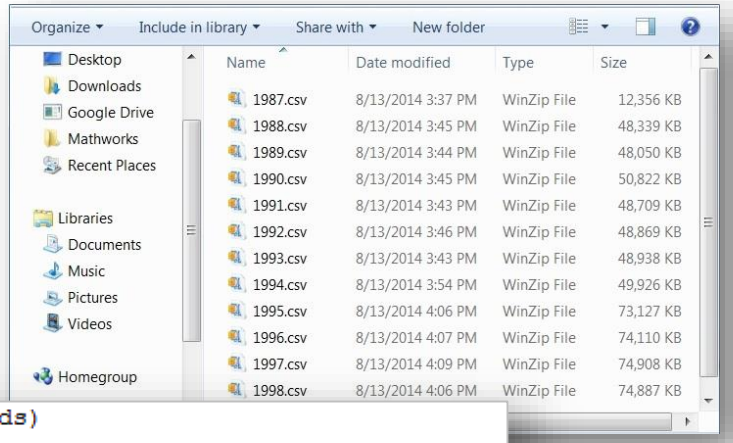
Access a subset of data from your desktop

Text Files local/network drive Hadoop	**Datastore**
Databases	**Datastore + Database Toolbox**
Binary Files local/network drive	**Memmap**

Access Big Data

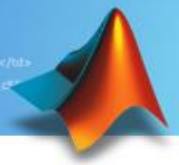
`datastore`

- Easily specify data set
  - Single text file (or collection of text files)
  - Database (using Database Toolbox)
- Preview data structure and format
- Select data to import using column names
- Incrementally read subsets of the data




```
>> preview(ds)
ans =
    Year    Month    DayofMonth    DayOfWeek
    ____    _____    __________    _________
    1987    10       21            3
    1987    10       26            1
    1987    10       23            5
    1987    10       23            5
```

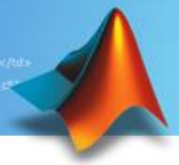
```
airdata = datastore('*.csv');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};

data = read(airdata);
```



How to use “DATASTORE” function

- `ds = datastore(location,Name,Value)`
  - Location:
    - file direction (‘C:\dir\data\file.csv’)
    - Read from HDFS (‘hdfs://myserver:7867/data/file1.txt’)
  - Name-Value Pair
    - TextscanFormats
    - SelectedVariableNames, SelectedFormat
    - Delimiter
    - ReadSize

- Methods of DATASTORE object
  - preview, read, readall



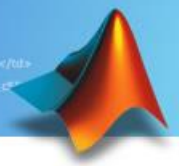
Analyze Big Data

mapreduce

- Use the powerful MapReduce programming technique to analyze big data
  - Multiple items (keys) to organize and process
  - Intermediate results do not fit in memory
- On the desktop
  - Analyze big database tables (Database Toolbox)
  - Increase compute capacity (Parallel Computing Toolbox)
  - Access data on HDFS to develop algorithms for use on Hadoop
- With Hadoop
  - Run on Hadoop using MATLAB Distributed Computing Server
  - Deploy applications and libraries for Hadoop using MATLAB Compiler

```
********************************
*      MAPREDUCE PROGRESS      *
********************************
Map 0%          Reduce 0%
Map 20%         Reduce 0%
Map 40%         Reduce 0%
Map 60%         Reduce 0%
Map 80%         Reduce 0%
Map 100%        Reduce 25%
Map 100%        Reduce 50%
Map 100%        Reduce 75%
Map 100%        Reduce 100%
```

Mapreduce (update for car reg demo)



mapreduce

Data Store

Veh_typ	Q3_08	Q4_08	Q1_09	Hybrid
Car	1	1	1	0
SUV	0	1	1	1
Car	1	1	1	1
Car	0	0	1	1
Car	0	1	1	1
Car	1	1	1	1
Car	0	0	1	1
SUV	0	1	1	0
Car	1	1	1	0
SUV	1	1	1	1
Car	0	1	1	1
Car	1	0	0	0

Map

Hybrid	
0	Key: Q3_08
1	
1	
0	Key: Q4_08
1	
1	
1	
0	Key: Q1_09
1	
1	
1	
0	Key: Q3_08
0	
0	Key: Q4_08
1	
0	Key: Q1_09
1	

Shuffle and Sort

Hybrid	
0	Key: Q3_08
1	
1	
0	
0	
0	Key: Q4_08
1	
1	
1	
0	
1	
0	Key: Q1_09
1	
1	
1	
1	
0	
1	

Reduce

Key	% Hybrid (Value)
Q3_08	0.4
Q4_08	0.67
Q1_09	0.75

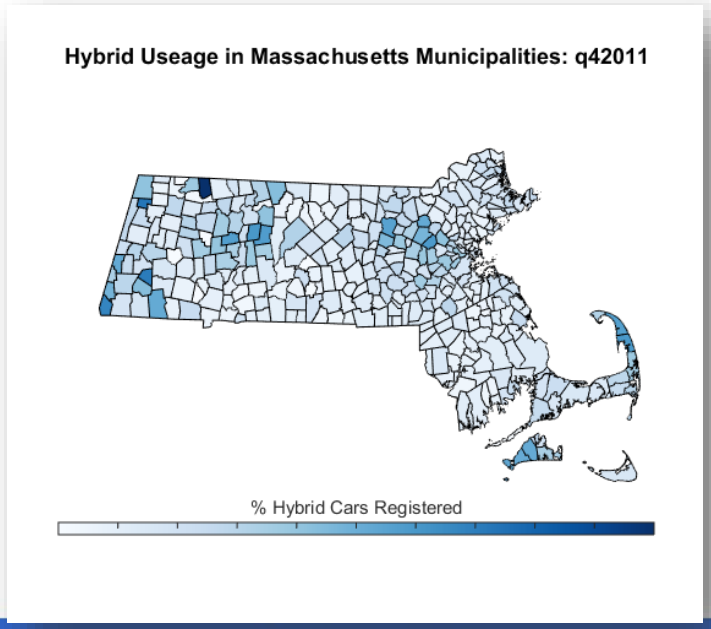
Demo: Vehicle Registry Analysis

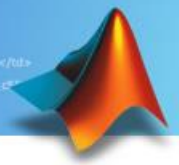
Using MapReduce

- Data
  - Massachusetts Vehicle Registration Data from 2008-2011
  - 16M records, 45 fields

muni_id	veh_zip	insp_year	model_year	make
325	1089	2011	2008	'Hyundai'
325	1089	2009	2008	'Hyundai'
288	1776	2011	2008	'Acura'
288	1776	2008	2008	'Acura'
145	2364	2011	2005	'Chevrolet'
325	1089	2010	2008	'Hyundai'
325	1089	2011	2008	'Hyundai'
288	1776	2009	2008	'Acura'

- Analysis
  - Examine hybrid adoptions
  - Calculate % of hybrids registered
    - By Quarter
    - By Regional Area
  - Create map of results





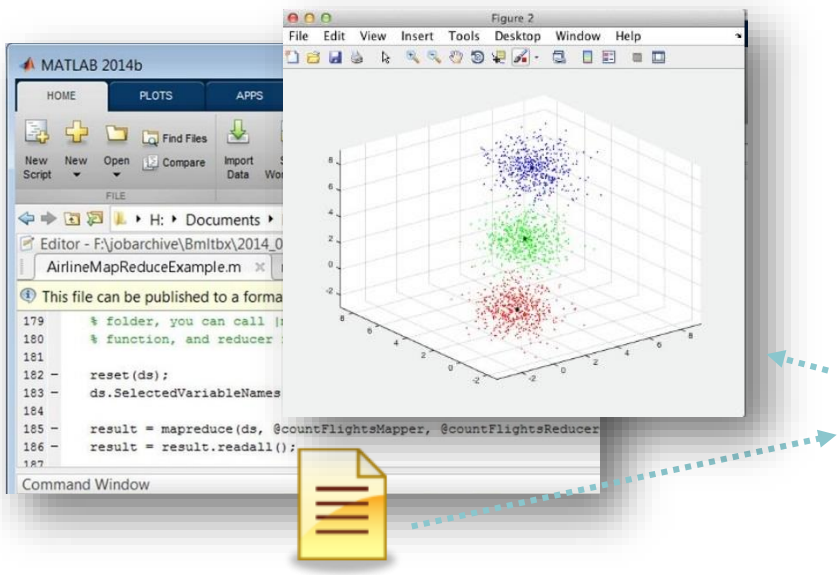
MATLAB on Hadoop

Two modes of operation

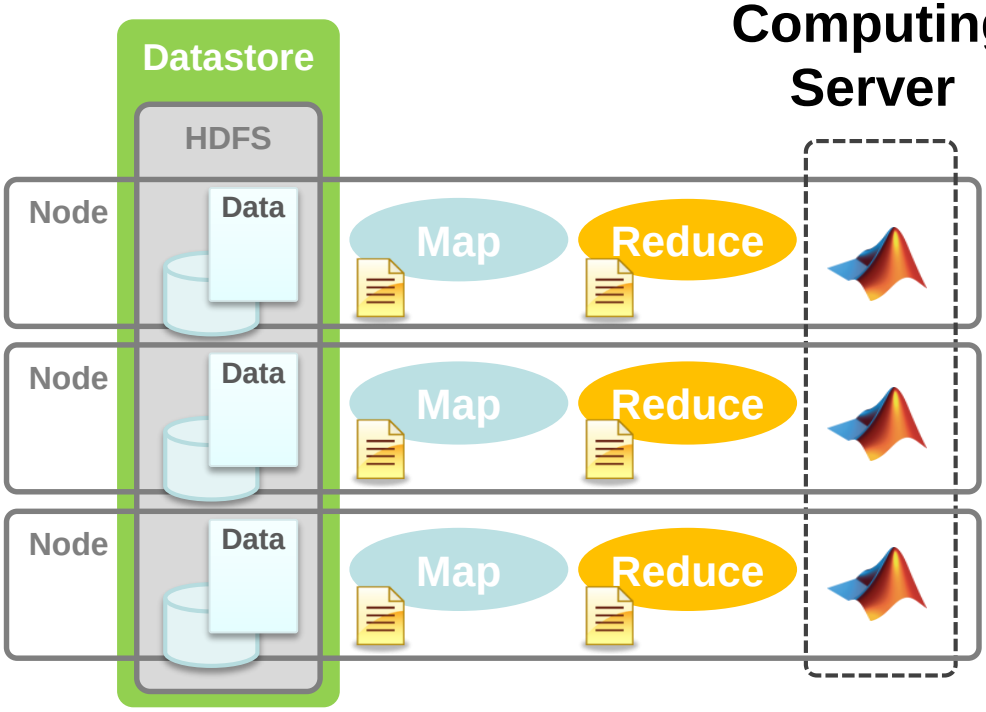
- Execute `mapreduce` on Hadoop from your MATLAB desktop using *MATLAB Distributed Computing Server*
  - Extends your desktop environment for use with Hadoop
  - Execute algorithms within Hadoop MapReduce on data stored in HDFS
- Create standalone applications or libraries for deploying to production instances of Hadoop
  - Locked down package for use in production environments
  - Integration of MATLAB analytics with operational systems

MATLAB Distributed Computing Server

with Hadoop



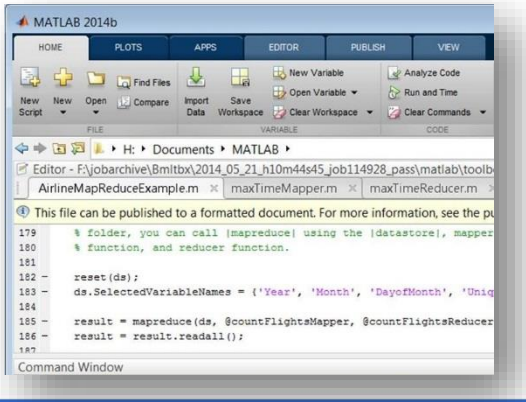
MATLAB MapReduce Code



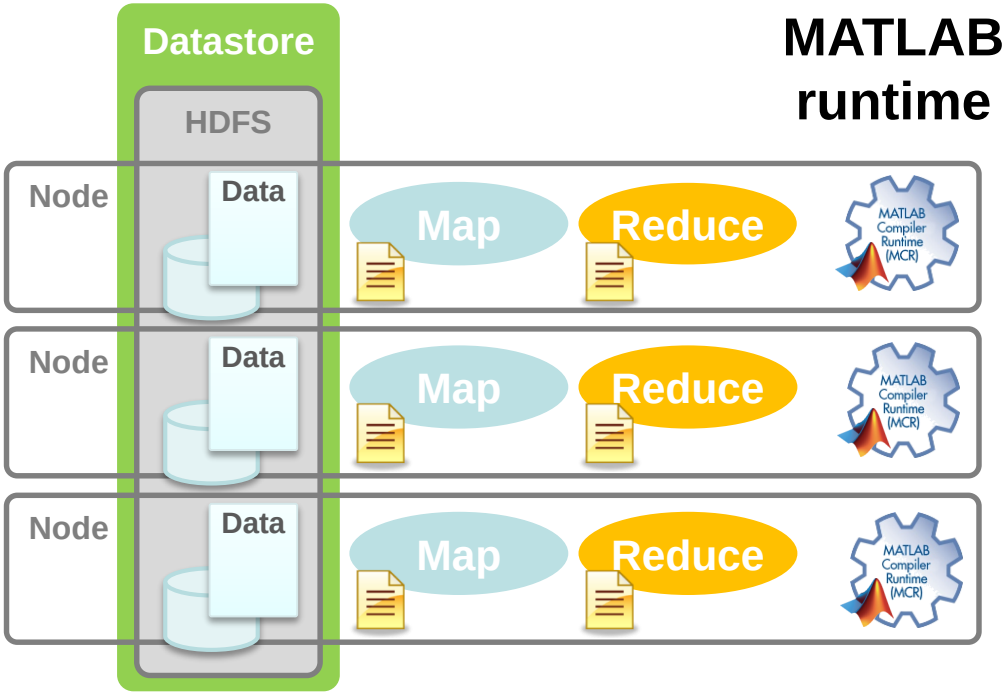
Hadoop

MATLAB Compiler

with Hadoop



MATLAB MapReduce Code

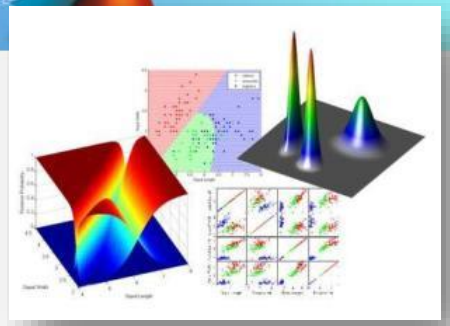


MATLAB runtime

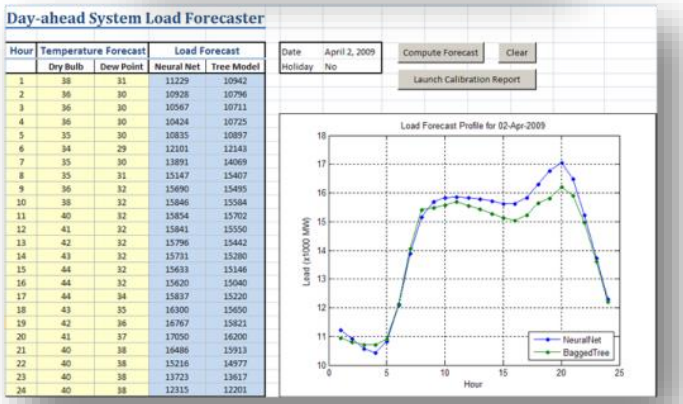
Hadoop

Machine Learning

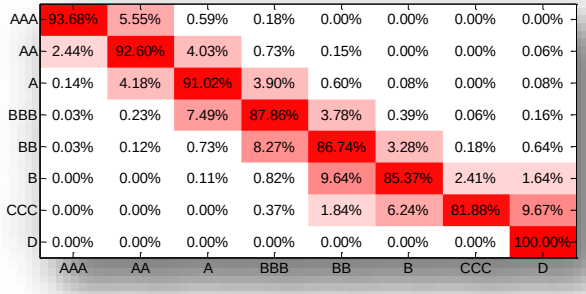
Characteristics and Examples

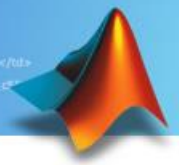


- Characteristics
  - Lots of data (many variables)
  - System too complex to know the governing equation *(e.g., black-box modeling)*



- Examples
  - Pattern recognition *(speech, images)*
  - Financial algorithms *(credit scoring, algo trading)*
  - Energy forecasting *(load, price)*
  - Biology *(tumor detection, drug discovery)*



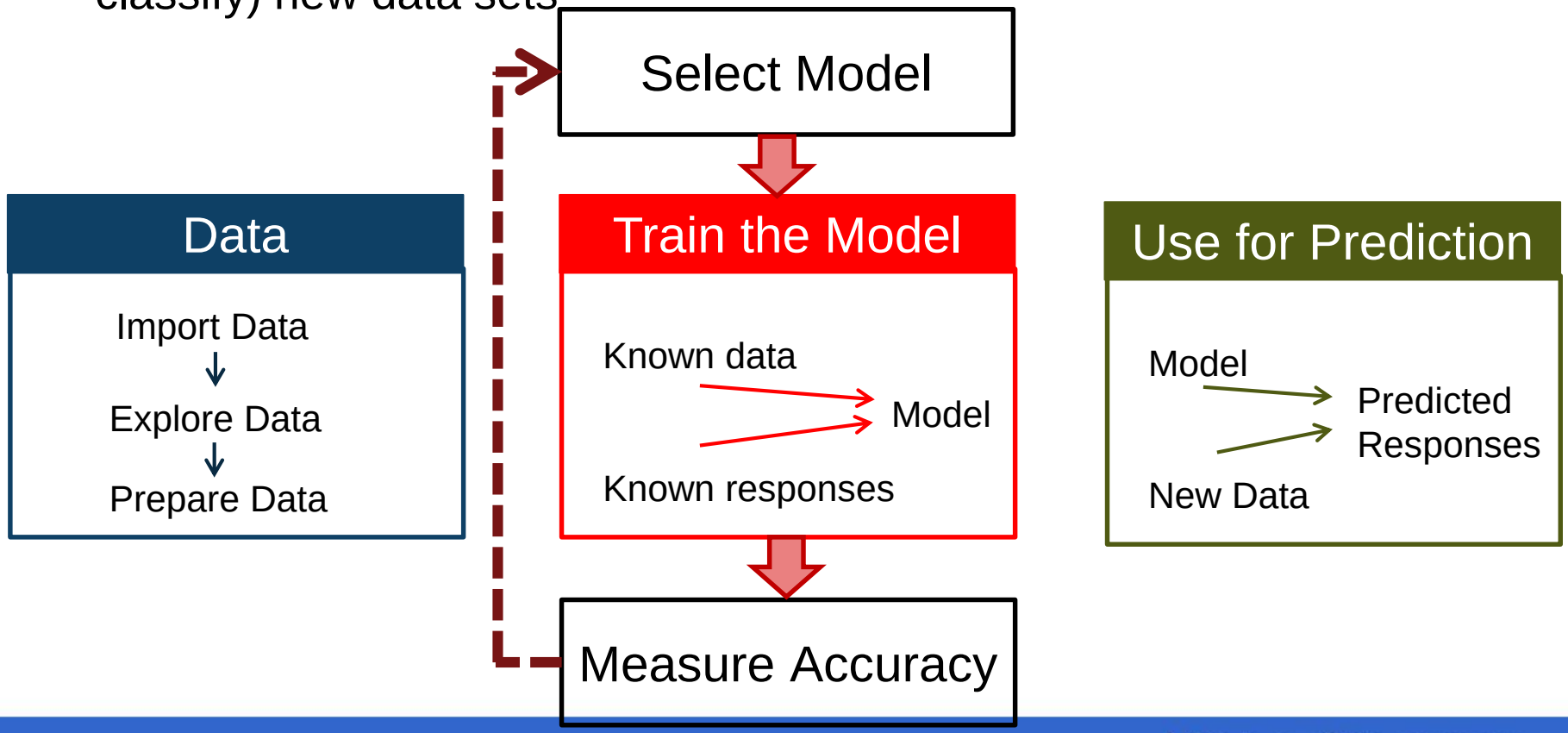


Challenges – Machine Learning

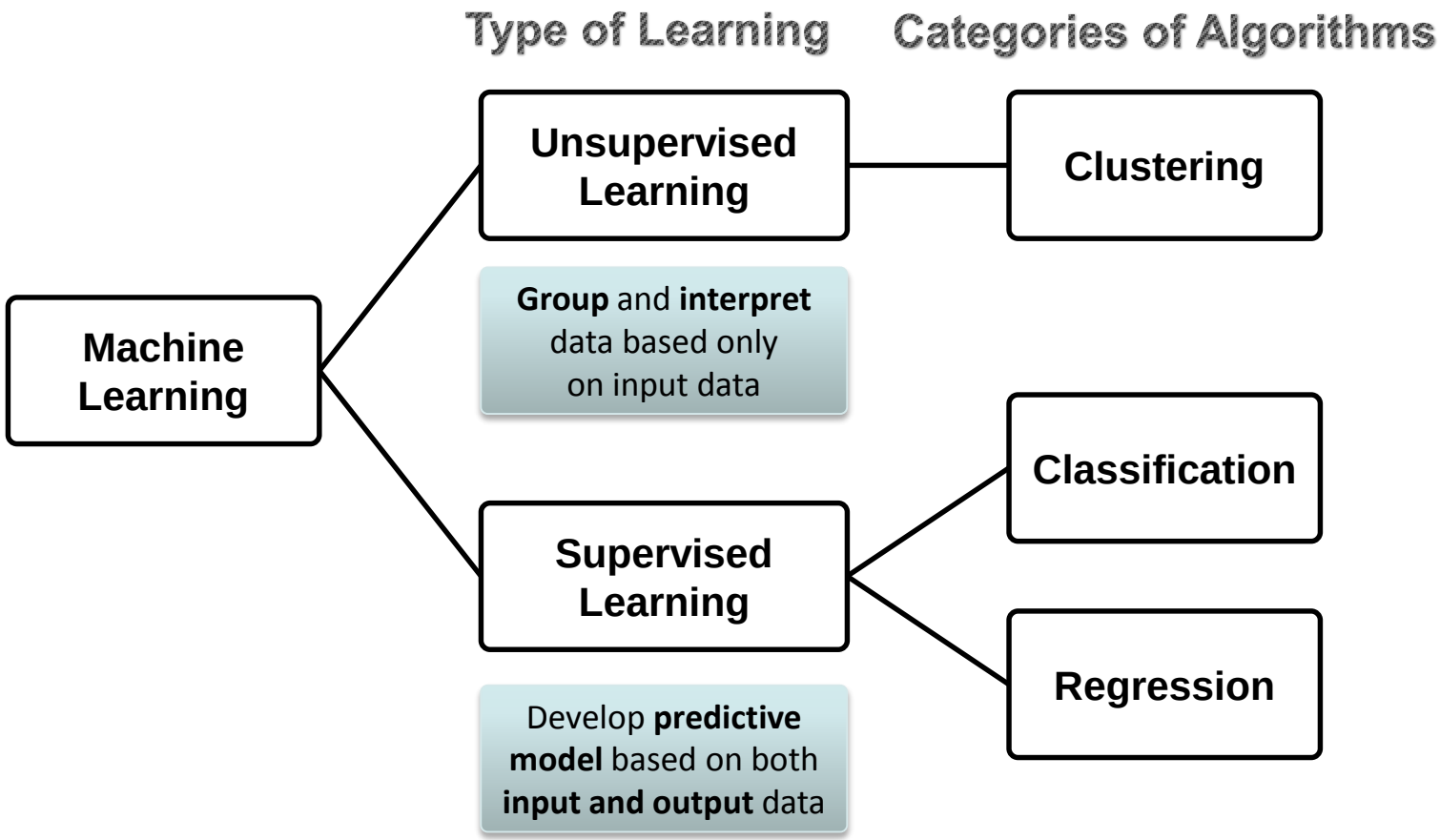
- Significant technical expertise required
- No “one size fits all” solution
- Locked into Black Box solutions
- Time required to conduct the analysis

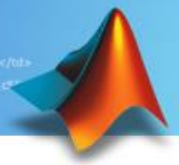
Why is it called “learning”?

- “Train” algorithm with initial data
- Use resulting model (or knowledge) to predict outcomes for (or classify) new data sets

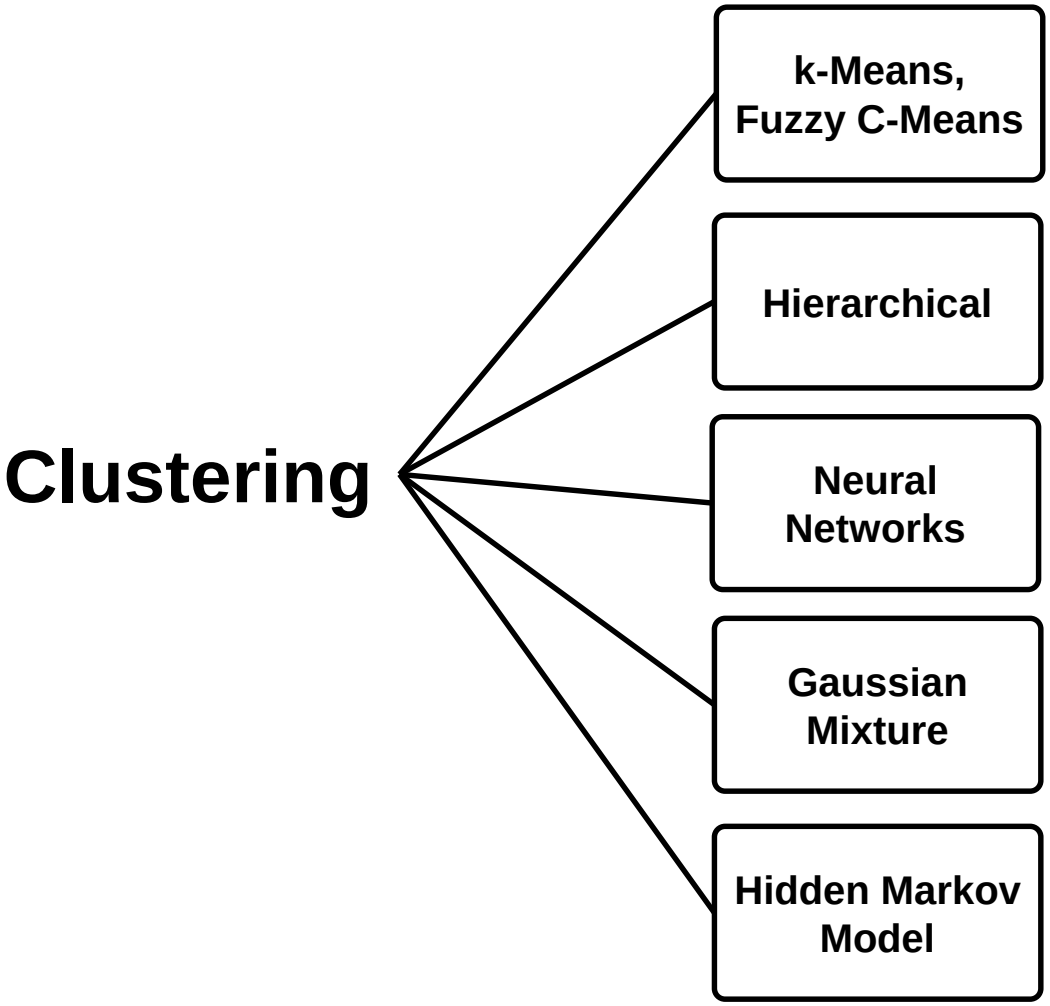


Overview – Machine Learning

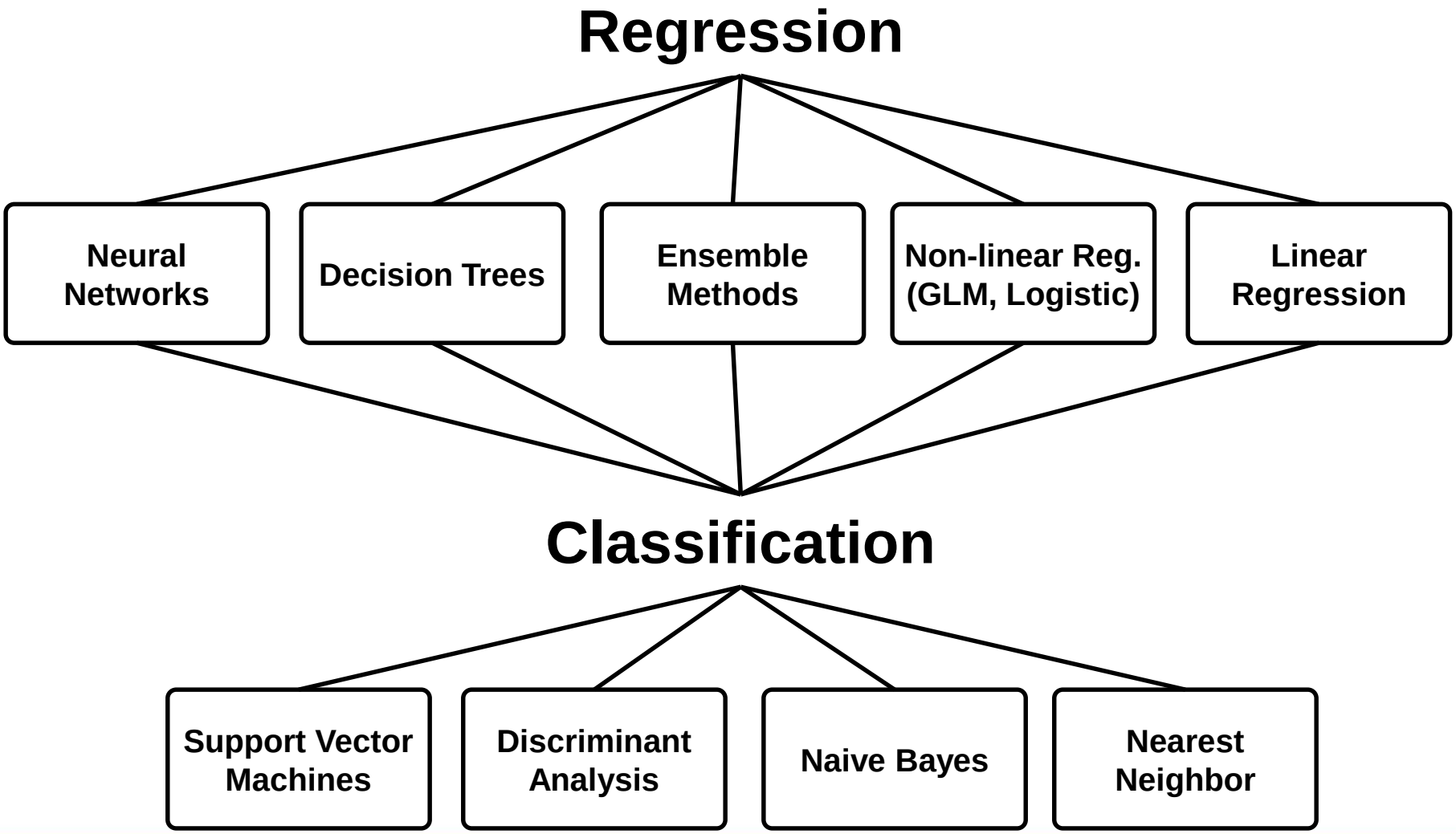




Unsupervised Learning

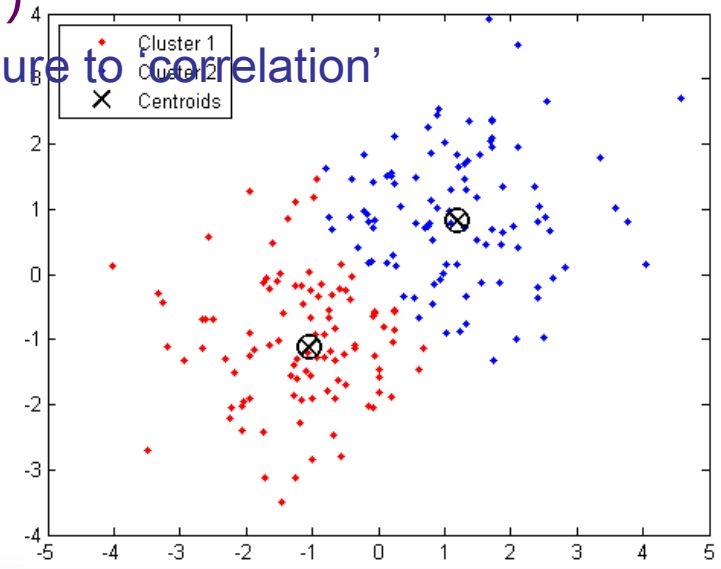


Supervised Learning



Clustering : k-means

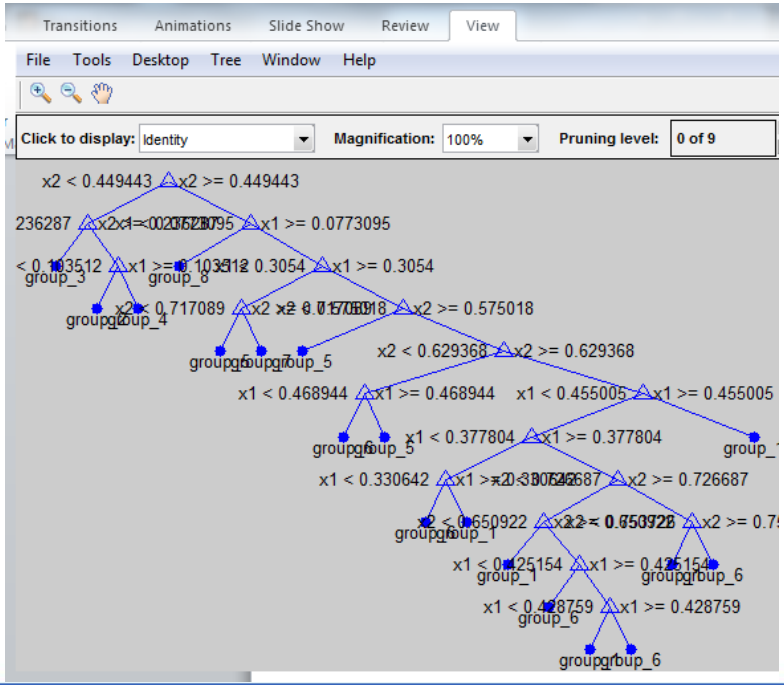
- Clustering by k-means
  - kmeans(Inputs, k)
    - Inputs : Rows of Inputs correspond to points, columns correspond to variables
    - k : number of cluster centroid locations
  - kmeans(Inputs, k, 'distance', 'correlation')
    - The parameter change the distance measure to 'correlation'




```
>> load kmeansdata
```

Classification : Classification tree

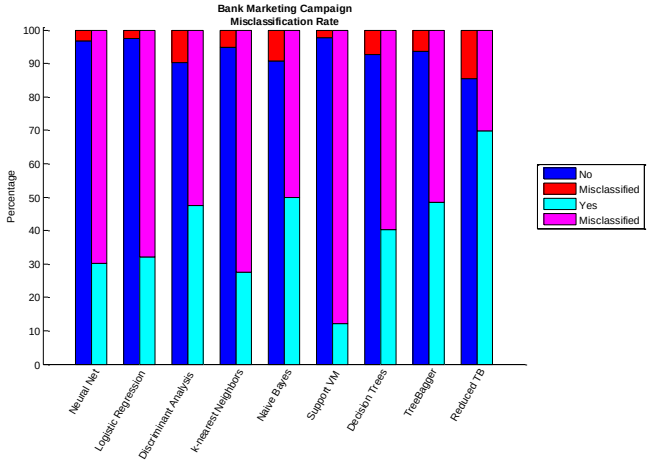
- Train a tree
  - tree = ClassificationTree.fit(x, y);
    - x : predictor values
    - y : Each row of y represents the classification of the corresponding row of x.
- View a classification tree
  - view(tree)
  - view(tree, 'mode', 'graph')
- Predict by the trained tree
  - Ypred = predict(tree, newX)



>> load ionosphere

Example – Bank Marketing Campaign

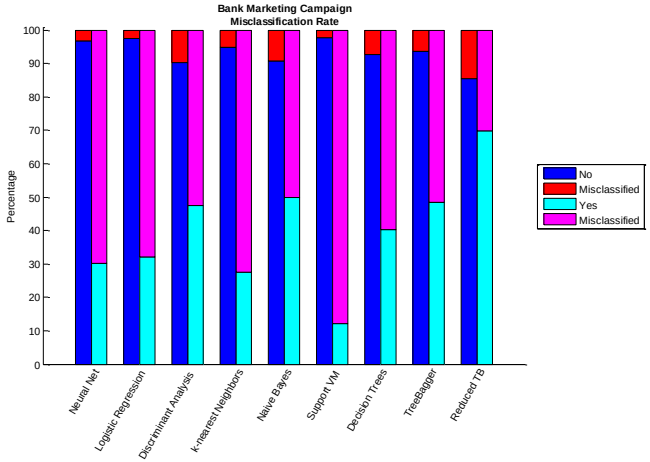
- Goal:
  - Predict if customer would subscribe to bank term deposit based on different attributes
- Approach:
  - Train a classifier using different models
  - Measure accuracy and compare models
  - Reduce model complexity
  - Use classifier for prediction



Data set downloaded from UCI Machine Learning repository
http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

Example – Bank Marketing Campaign

- Numerous predictive models with rich documentation
- Interactive visualizations and apps to aid discovery
- Built-in parallel computing support
- Quick prototyping; Focus on modeling not programming

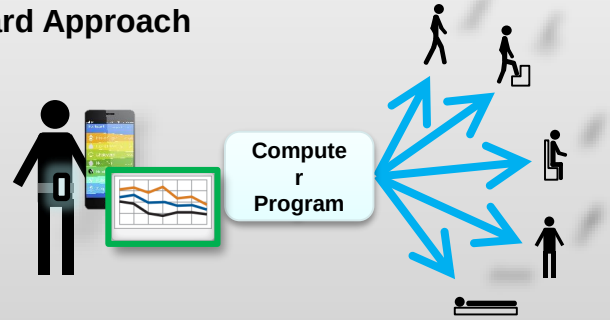


Machine Learning

Machine learning uses **data** and produces a **program** to perform a **task**

Task: Human Activity Detection

Standard Approach



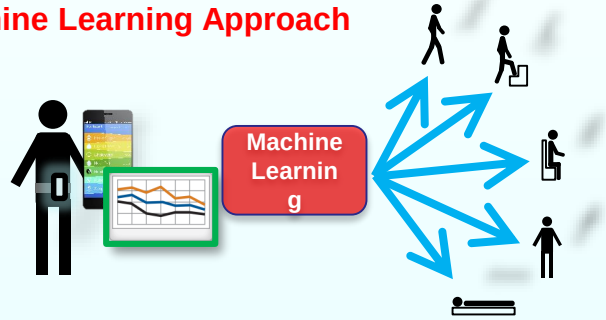
Hand Written Program

If X_acc > 0.5
then “SITTING”
If Y_acc < 4 and Z_acc > 5
then “STANDING”
…

Formula or Equation

$$Y_{activity} = \beta_1 X_{acc} + \beta_2 Y_{acc} + \beta_3 Z_{acc} + \dots$$

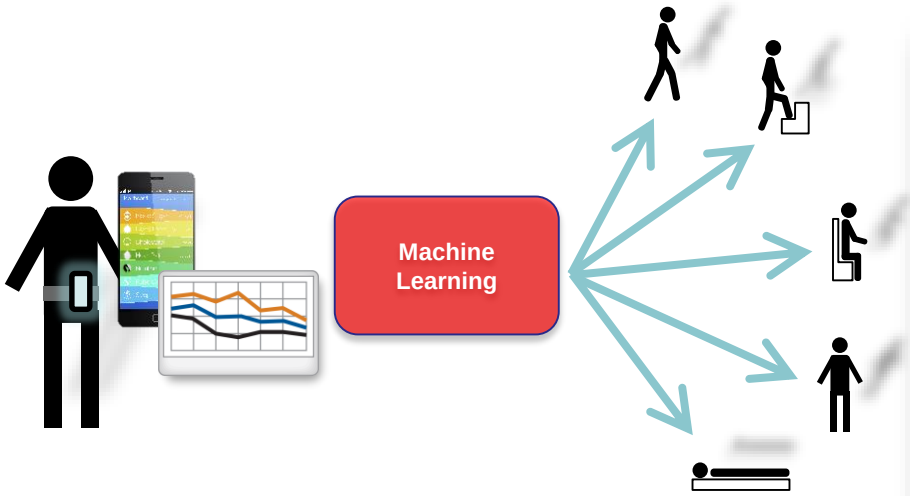
Machine Learning Approach

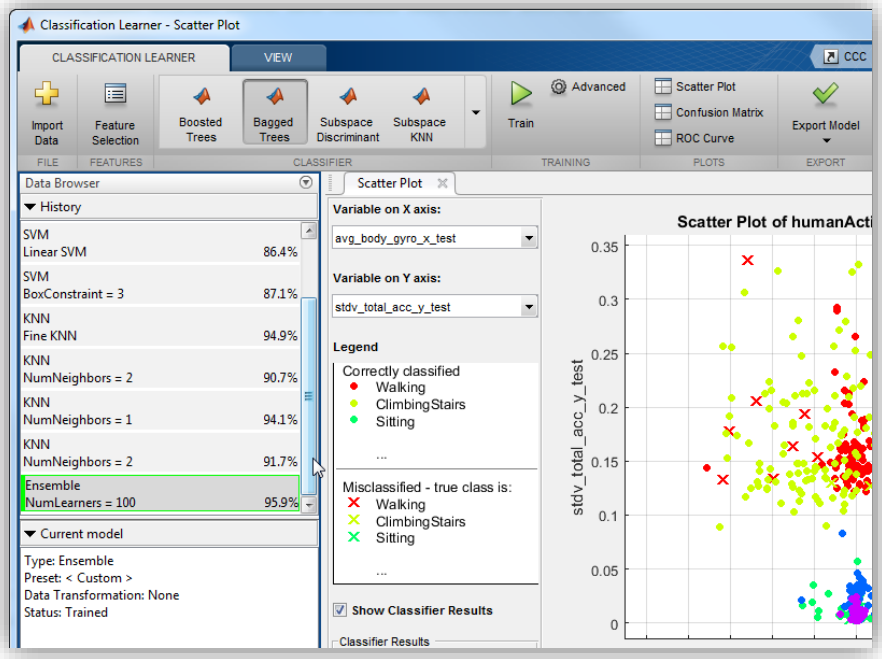


$model$: Inputs → Outputs

$$model = <^{Machine\ Learning\ Algorithm}>(sensor\_data, activity)$$

Example: Human Activity Learning Using Mobile Phone Data





Data:

- 3-axial Accelerometer data
- 3-axial Gyroscope data

“essentially, all models are wrong, but some are useful”

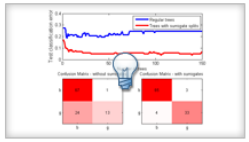
– George Box

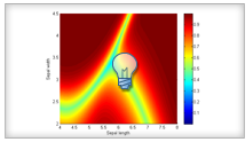
Learn More: Machine Learning with MATLAB

mathworks.com/machine-learning

Classification Examples

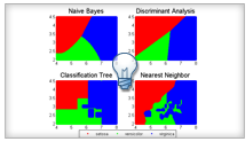

Basket Selection Using Stepwise Regression


Classification in the Presence of Missing Data

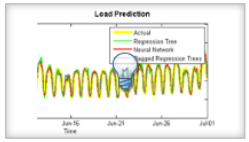

Classification Probability

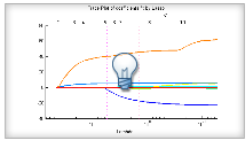

Digit Classification Using HOG Features

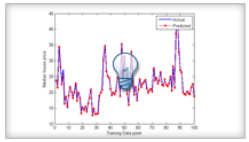

Handwriting Recognition Using Bagged Classification Trees


Visualize Decision Surfaces for Different Classifiers

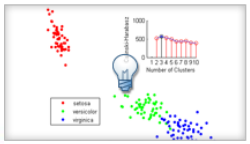
Regression Examples


Electricity Load Forecasting


Lasso Regularization


Regression with Boosted Decision Trees

Clustering Examples


Cluster Evaluation


Cluster Genes Using K-Means and Self-Organizing Maps


Color-Based Segmentation Using K-Means Clustering