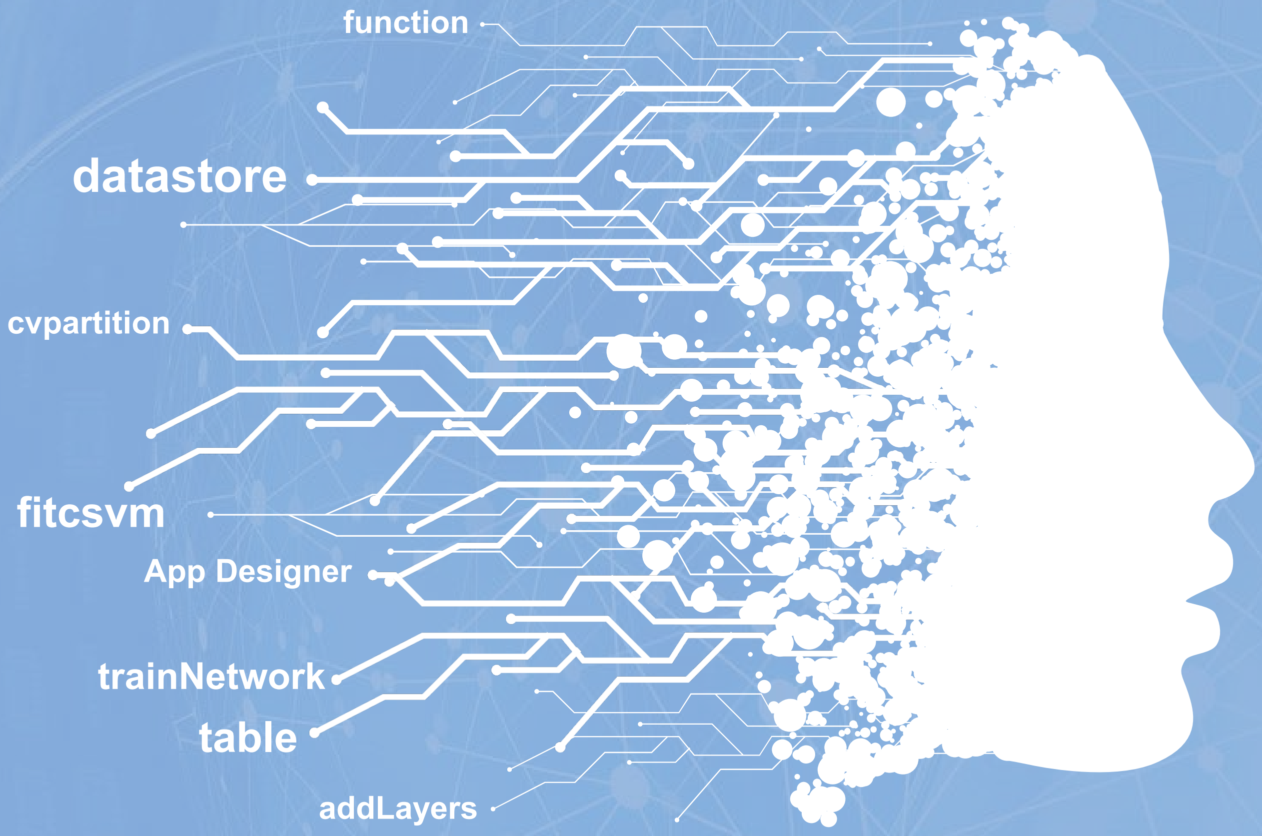
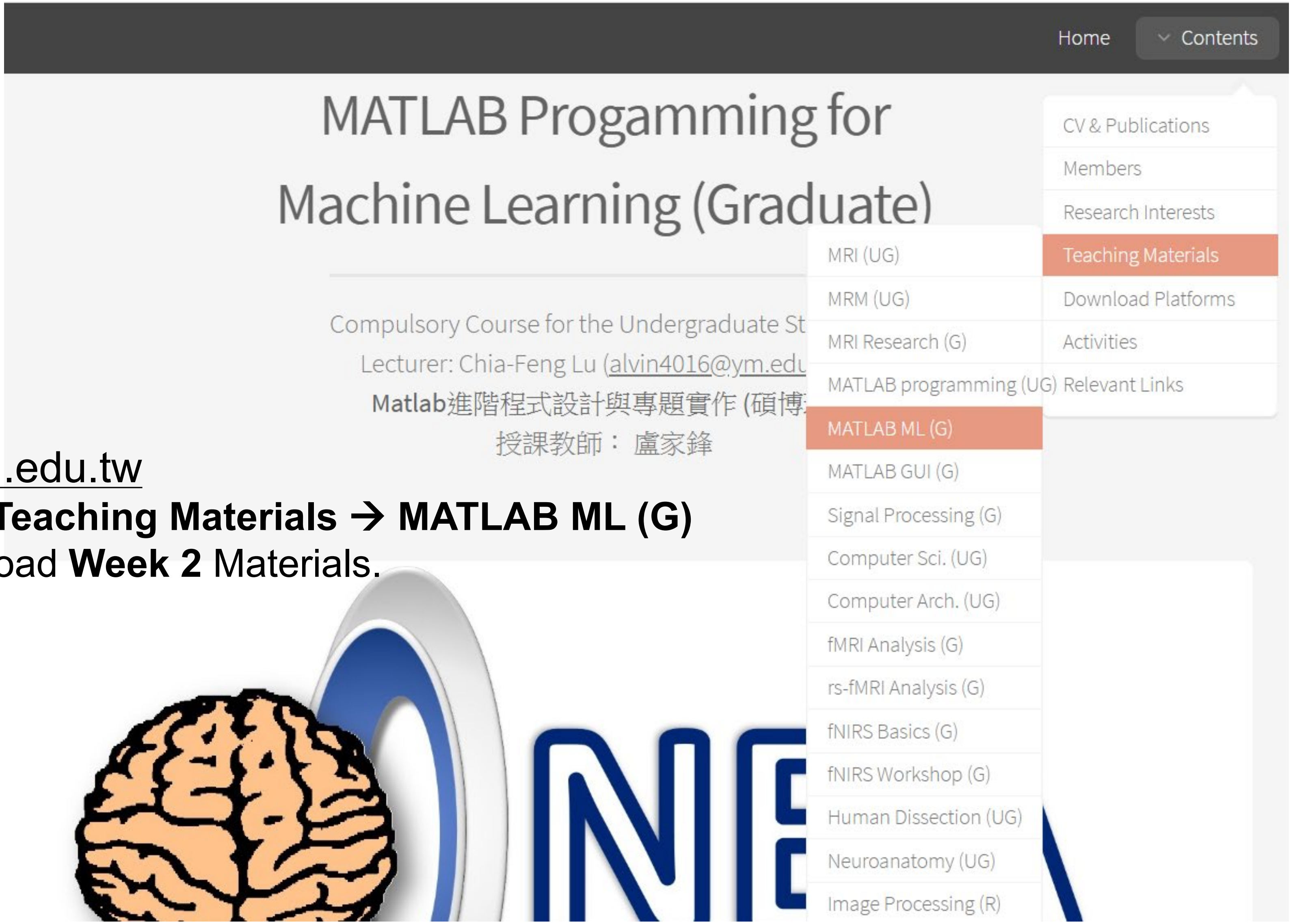


Introduction of Machine Learning

MATLAB進階程式語言與實作

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging and Radiological Sciences, NYCU
alvin4016@nycu.edu.tw

Teaching Materials



cflu.lab.nycu.edu.tw

Contents → Teaching Materials → MATLAB ML (G)

Please download **Week 2** Materials.

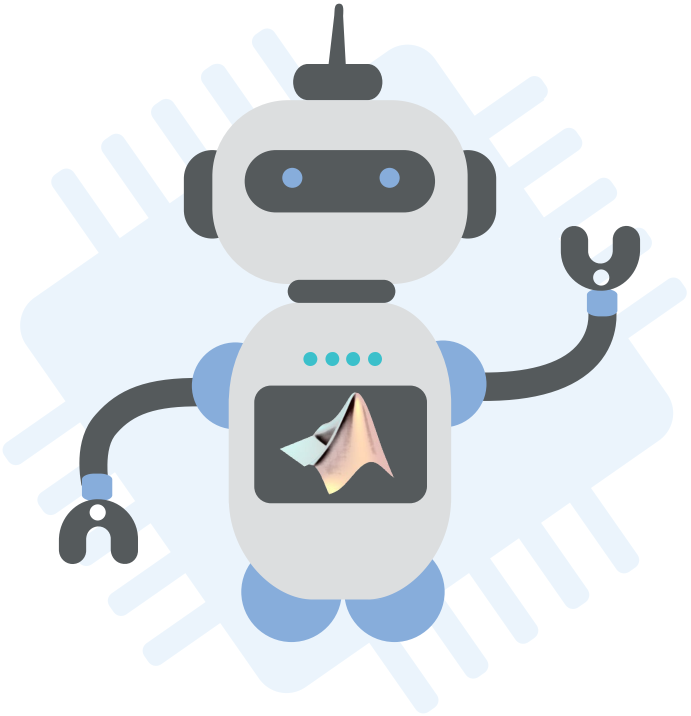
Contents in this Week

01 General Concepts of Machine Learning

Purpose, key components, applications, and pitfalls/limitations

02 Machine Learning Workflow

A step-by-step example of the MATLAB ML application



References & Resources

- Section 1: Introducing Machine Learning
- Section 2: Getting Started with Machine Learning
- Section 3: Applying Unsupervised Learning
- Section 4: Applying Supervised Learning

- MATLAB Machine Learning Examples

mathworks.com/campaigns/offers/machine-learning-with-matlab.confirmation.html?elqsid=1583760350878&potential_use=Education

General Concepts of Machine Learning

Purpose, key components, applications, and pitfalls/limitations

What is Machine Learning?

- Machine learning teaches computers to do what comes naturally to humans and animals: learn from experience.

The key features of a CAR?
(Compared to a motorcycle)

What is Machine Learning (ML)?

- Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model.

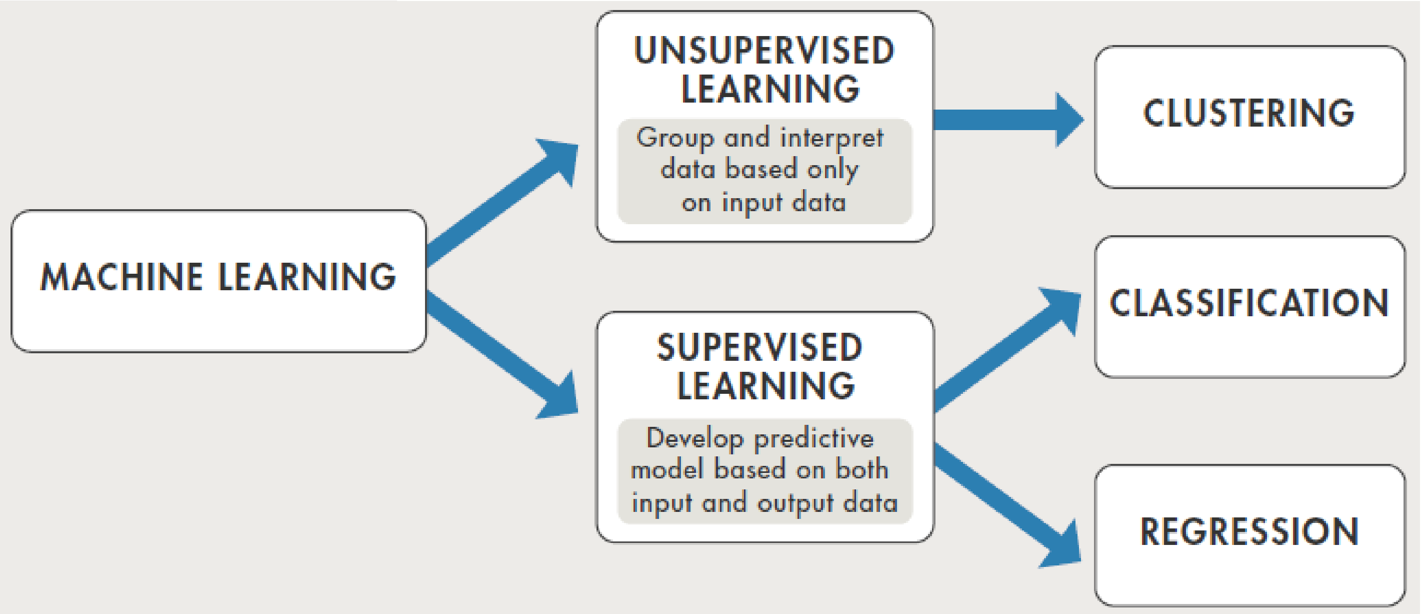
- The algorithms adaptively improve their performance as the number of samples available for learning increases.

$$y = 3.2x_1^2 + 2.6x_2^2 + 1.2x_1x_2$$

$$\begin{cases} m \geq 4.23, y = 1\ (infected) \\ m < 4.23, y = 0\ (healthy) \end{cases}$$

Types of ML

- **Unsupervised learning**
  - I am self-sufficient in learning.
- **Supervised learning**
  - Train Me!
- **Reinforcement learning**
  - My life My rules! (Hit & Trial)

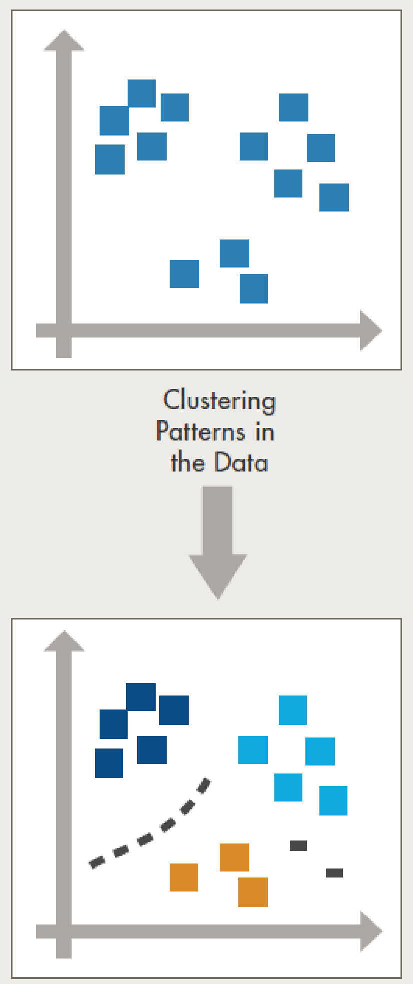


Machine Learning with MATLAB, Section 1

Unsupervised Learning

- Drawing inferences from datasets consisting of input data without labeled responses.

- **Clustering** is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data.
- Applications for clustering include gene sequence analysis, market research, and object recognition.



Machine Learning with MATLAB, Section 1

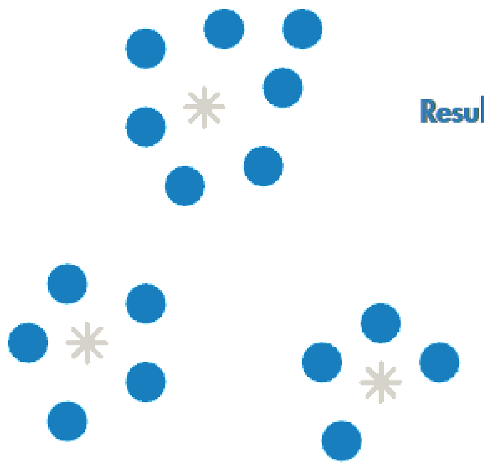
Hard Clustering Algorithms

k-Means

How it Works
Partitions data into k number of mutually exclusive clusters. How well a point fits into a cluster is determined by the distance from that point to the cluster's center.

Best Used...

- When the number of clusters is known
- For fast clustering of large data sets



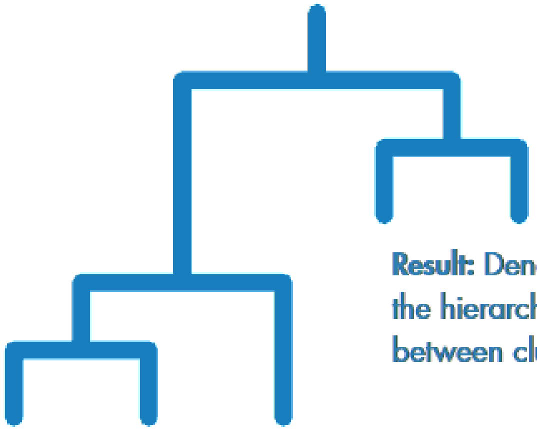
Result: Cluster centers

Hierarchical Clustering

How it Works
Produces nested sets of clusters by analyzing similarities between pairs of points and grouping objects into a binary, hierarchical tree.

Best Used...

- When you don't know in advance how many clusters are in your data
- You want visualization to guide your selection



Result: Dendrogram showing the hierarchical relationship between clusters

Machine Learning with MATLAB, Section 3

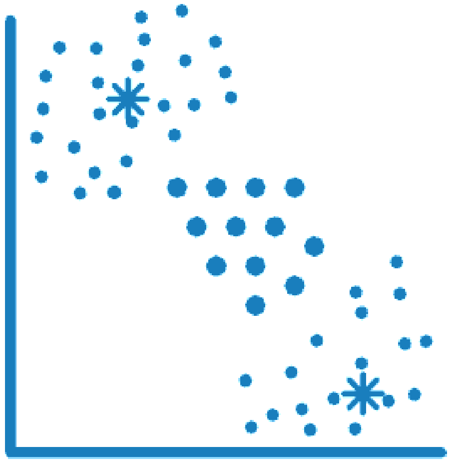
Soft Clustering Algorithms

Fuzzy c-Means

How it Works
Partition-based clustering when data points may belong to more than one cluster.

Best Used...

- When the number of clusters is known
- For pattern recognition
- When clusters overlap



Result: Cluster centers (similar to k-means) but with fuzziness so that points may belong to more than one cluster

Gaussian Mixture Model

How It Works
Partition-based clustering where data points come from different multivariate normal distributions with certain probabilities.

Best Used...

- When a data point might belong to more than one cluster
- When clusters have different sizes and correlation structures within them

Result: A model of Gaussian distributions that give probabilities of a point being in a cluster

Supervised Learning

- **Model training** requires…
  - A known set of input data
  - Known responses to the data (output)
- Application
  - To generate reasonable predictions for the response to new data.

- **Regression techniques** predict continuous responses (values)
  - For example, changes in temperature or fluctuations in power demand.
- **Classification techniques** predict discrete responses (categories)
  - For example, whether an email is genuine or spam, or whether a tumor is cancerous or benign.

Common Regression Algorithms

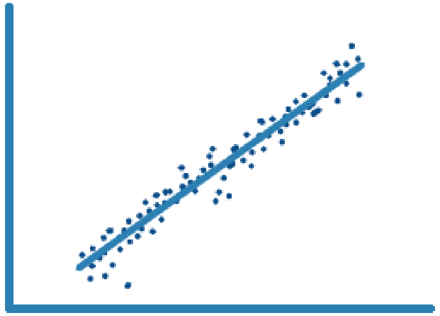
Linear Regression

How it Works

Linear regression is a statistical modeling technique used to describe a continuous response variable as a linear function of one or more predictor variables. Because linear regression models are simple to interpret and easy to train, they are often the first model to be fitted to a new dataset.

Best Used...

- When you need an algorithm that is easy to interpret and fast to fit
- As a baseline for evaluating other, more complex, regression models



Nonlinear Regression

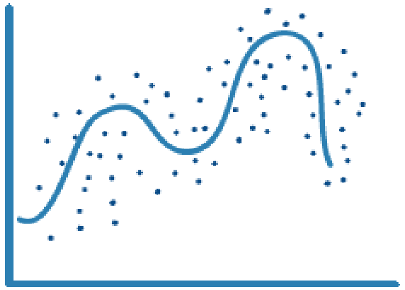
How It Works

Nonlinear regression is a statistical modeling technique that helps describe nonlinear relationships in experimental data. Nonlinear regression models are generally assumed to be parametric, where the model is described as a nonlinear equation.

"Nonlinear" refers to a fit function that is a nonlinear function of the parameters. For example, if the fitting parameters are b_0, b_1, and b_2: the equation $y = b_0+b_1x+b_2x^2$ is a linear function of the fitting parameters, whereas $y = (b_0x^{b_1})/(x+b_2)$ is a nonlinear function of the fitting parameters.

Best Used...

- When data has strong nonlinear trends and cannot be easily transformed into a linear space
- For fitting custom models to data



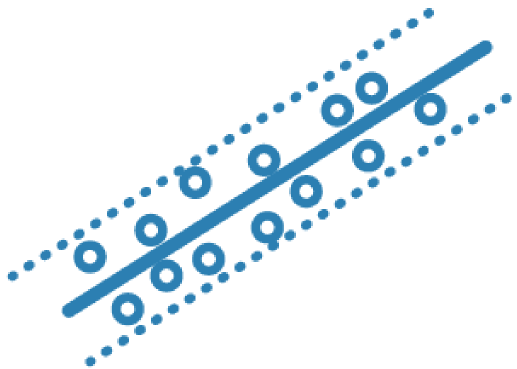
Common Regression Algorithms

SVM Regression

How It Works
SVM regression algorithms work like SVM classification algorithms, but are modified to be able to predict a continuous response. Instead of finding a hyperplane that separates data, SVM regression algorithms find a model that deviates from the measured data by a value no greater than a small amount, with parameter values that are as small as possible (to minimize sensitivity to error).

Best Used...

- For high-dimensional data (where there will be a large number of predictor variables)

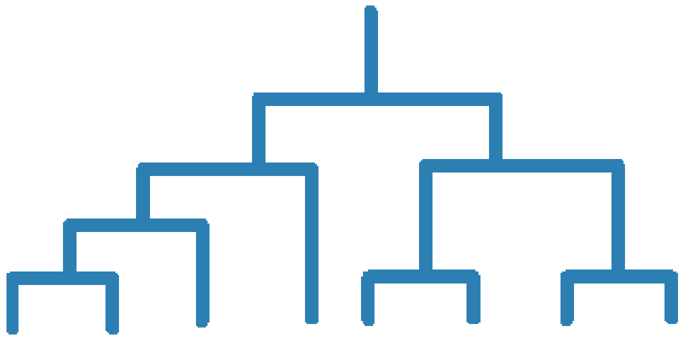


Regression Tree

How It Works
Decision trees for regression are similar to decision trees for classification, but they are modified to be able to predict continuous responses.

Best Used...

- When predictors are categorical (discrete) or behave nonlinearly



Common Classification Algorithms

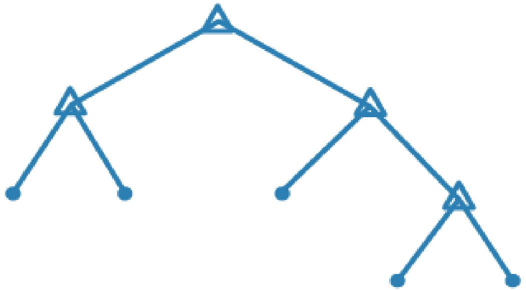
Decision Tree

How it Works

A decision tree lets you predict responses to data by following the decisions in the tree from the root (beginning) down to a leaf node. A tree consists of branching conditions where the value of a predictor is compared to a trained weight. The number of branches and the values of weights are determined in the training process. Additional modification, or pruning, may be used to simplify the model.

Best Used...

- When you need an algorithm that is easy to interpret and fast to fit
- To minimize memory usage
- When high predictive accuracy is not a requirement



Bagged and Boosted Decision Trees

How They Work

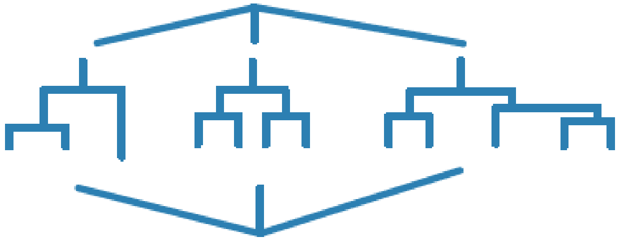
In these ensemble methods, several "weaker" decision trees are combined into a "stronger" ensemble.

A bagged decision tree consists of trees that are trained independently on data that is bootstrapped from the input data.

Boosting involves creating a strong learner by iteratively adding "weak" learners and adjusting the weight of each weak learner to focus on misclassified examples.

Best Used...

- When predictors are categorical (discrete) or behave nonlinearly
- When the time taken to train a model is less of a concern



Common Classification Algorithms

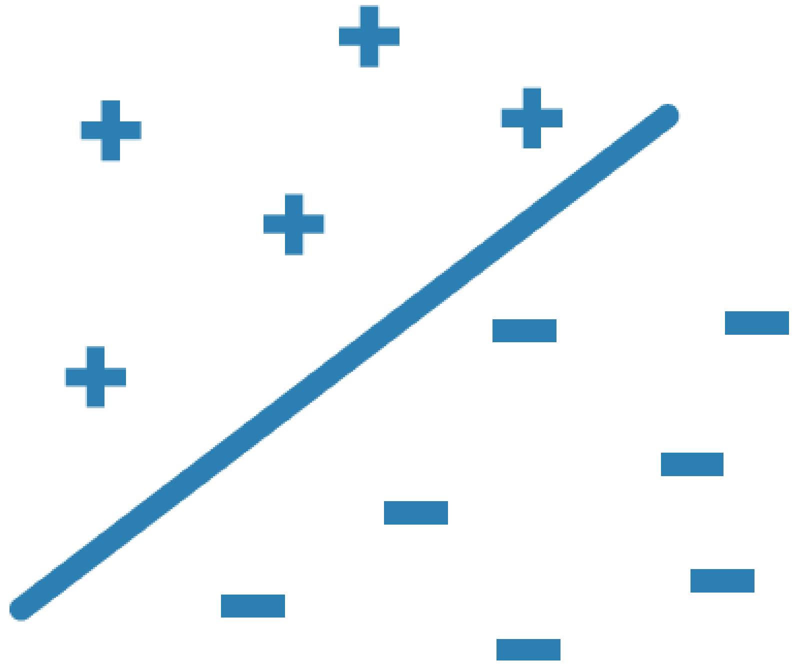
Support Vector Machine (SVM)

How It Works

Classifies data by finding the linear decision boundary (hyperplane) that separates all data points of one class from those of the other class. The best hyperplane for an SVM is the one with the largest margin between the two classes, when the data is linearly separable. If the data is not linearly separable, a loss function is used to penalize points on the wrong side of the hyperplane. SVMs sometimes use a kernel transform to transform nonlinearly separable data into higher dimensions where a linear decision boundary can be found.

Best Used...

- For data that has exactly two classes (you can also use it for multiclass classification with a technique called error-correcting output codes)
- For high-dimensional, nonlinearly separable data
- When you need a classifier that's simple, easy to interpret, and accurate



Common Classification Algorithms

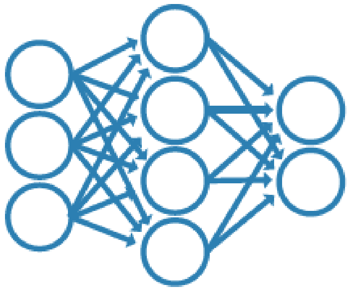
Neural Network

How it Works

Inspired by the human brain, a neural network consists of highly connected networks of neurons that relate the inputs to the desired outputs. The network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.

Best Used...

- For modeling highly nonlinear systems
- When data is available incrementally and you wish to constantly update the model
- When there could be unexpected changes in your input data
- When model interpretability is not a key concern



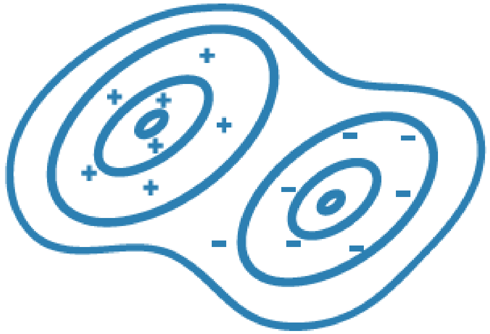
Naïve Bayes

How It Works

A naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It classifies new data based on the highest probability of its belonging to a particular class.

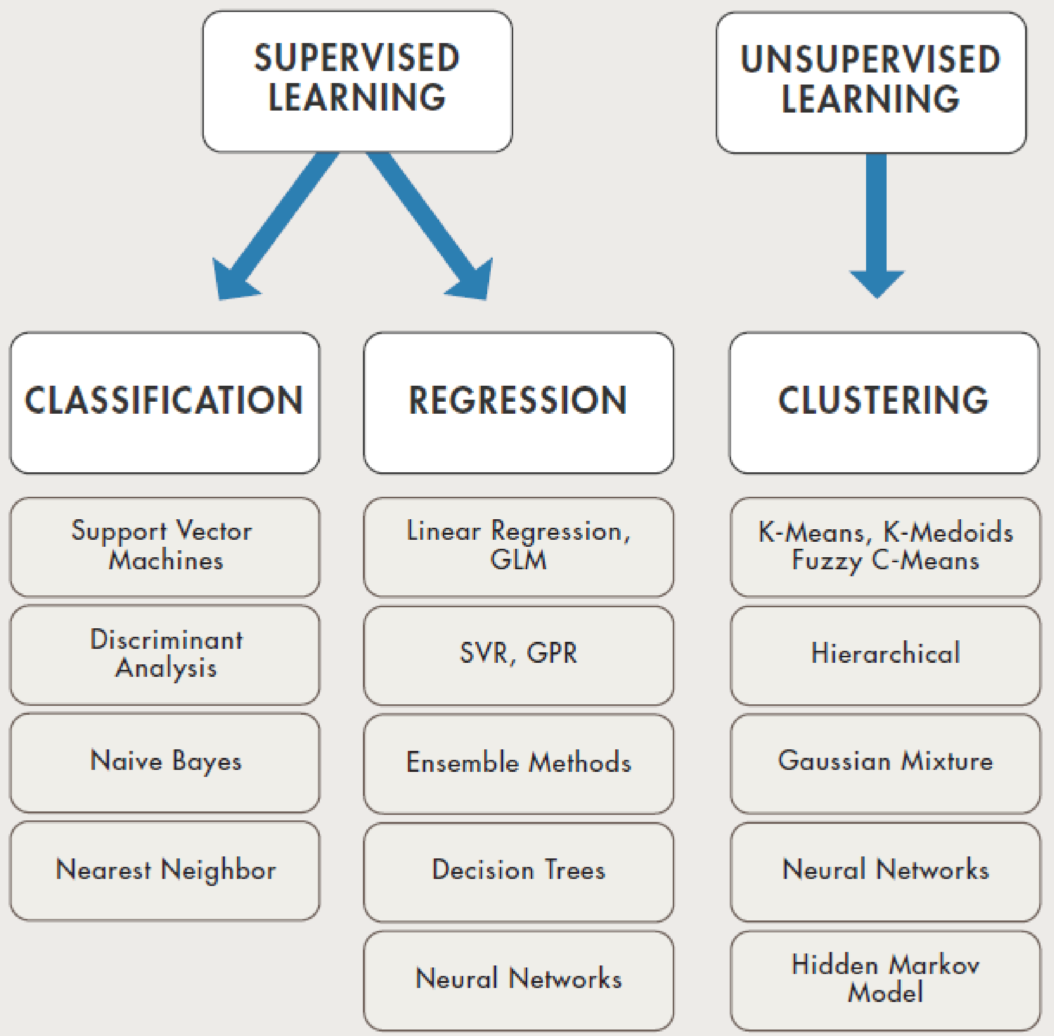
Best Used...

- For a small dataset containing many parameters
- When you need a classifier that's easy to interpret
- When the model will encounter scenarios that weren't in the training data, as is the case with many financial and medical applications



Selecting ML Algorithm

- There is no best method or one size fits all.
- Finding the right algorithm is partly just **trial and error**.
- Algorithm selection depends on…
  - the size and type of data you're working with,
  - the insights you want to get from the data,
  - how those insights will be used.

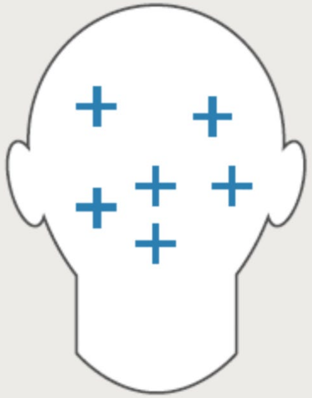


Machine Learning with MATLAB, Section 1

When Should You Use ML?

- When you have a complex task or problem involving a large amount of data and lots of variables, but no existing formula or equation.

Hand-written rules and equations are too complex—as in face recognition and speech recognition.



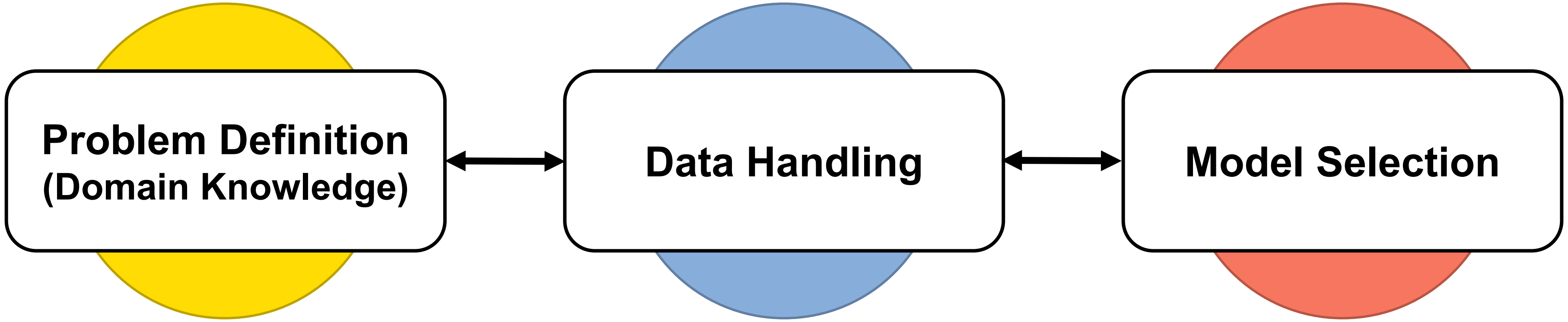
The rules of a task are constantly changing—as in fraud detection from transaction records.

The nature of the data keeps changing, and the program needs to adapt—as in automated trading, energy demand forecasting, and predicting shopping trends.

Machine Learning with MATLAB, Section 1

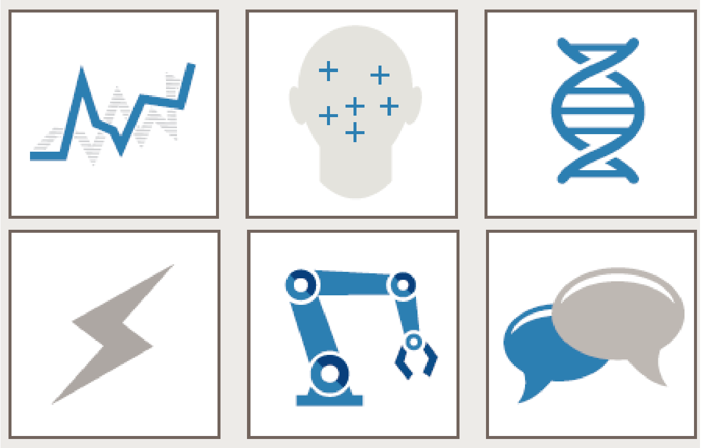
ML Challenges

- Data comes in all shapes and sizes.
  - Data format and variation
- Preprocessing your data might require specialized knowledge and tools.
  - Data preprocessing
- It takes time to find the best model to fit the data.
  - Experience & Trial-and-Error



Applications of ML

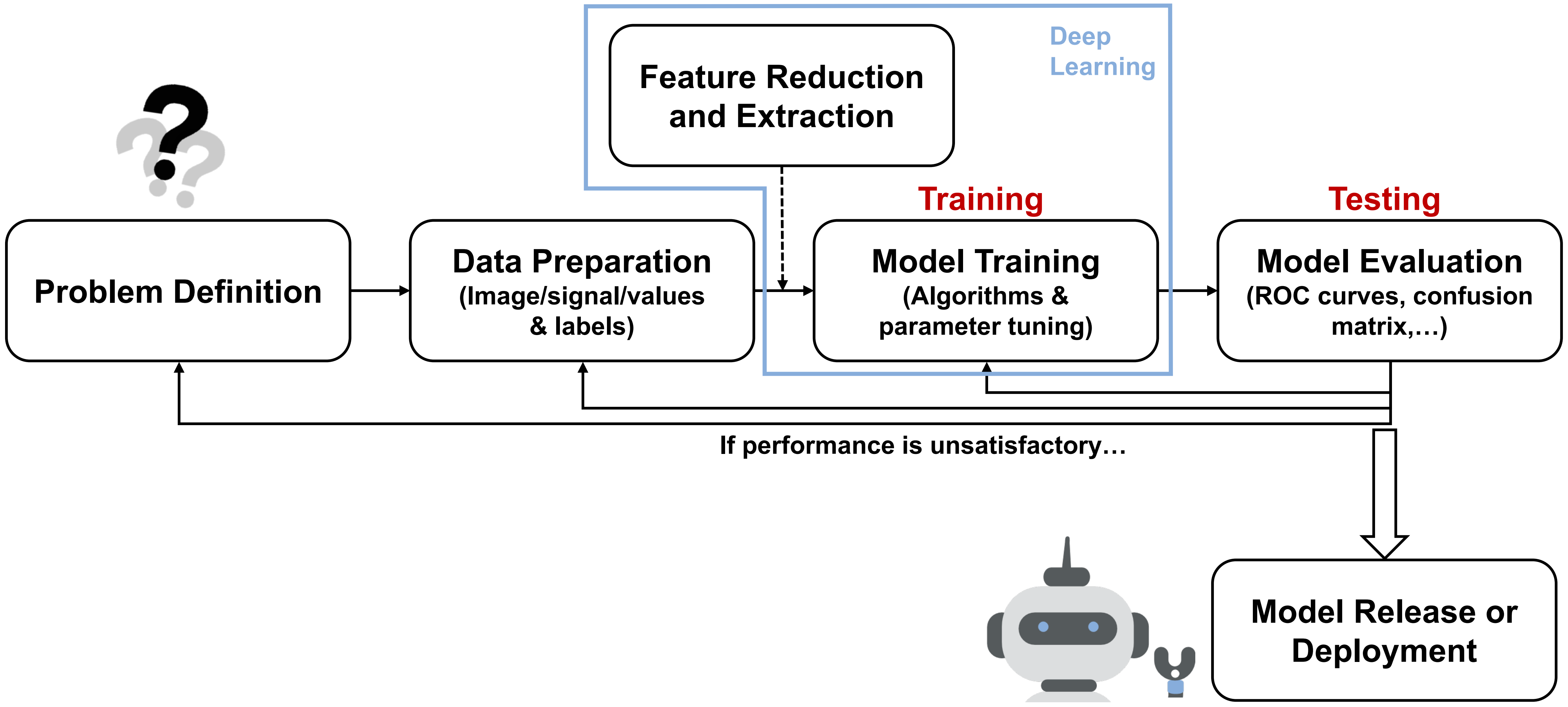
- Computational finance
  - Credit scoring and algorithmic trading
- Image processing and computer vision
  - Face recognition, motion detection, and object detection
- Computational biology
  - Tumor detection, drug discovery, and DNA sequencing
- Energy production
  - Price and load forecasting
- Automotive, aerospace, and manufacturing,
  - Predictive maintenance
- Natural language processing



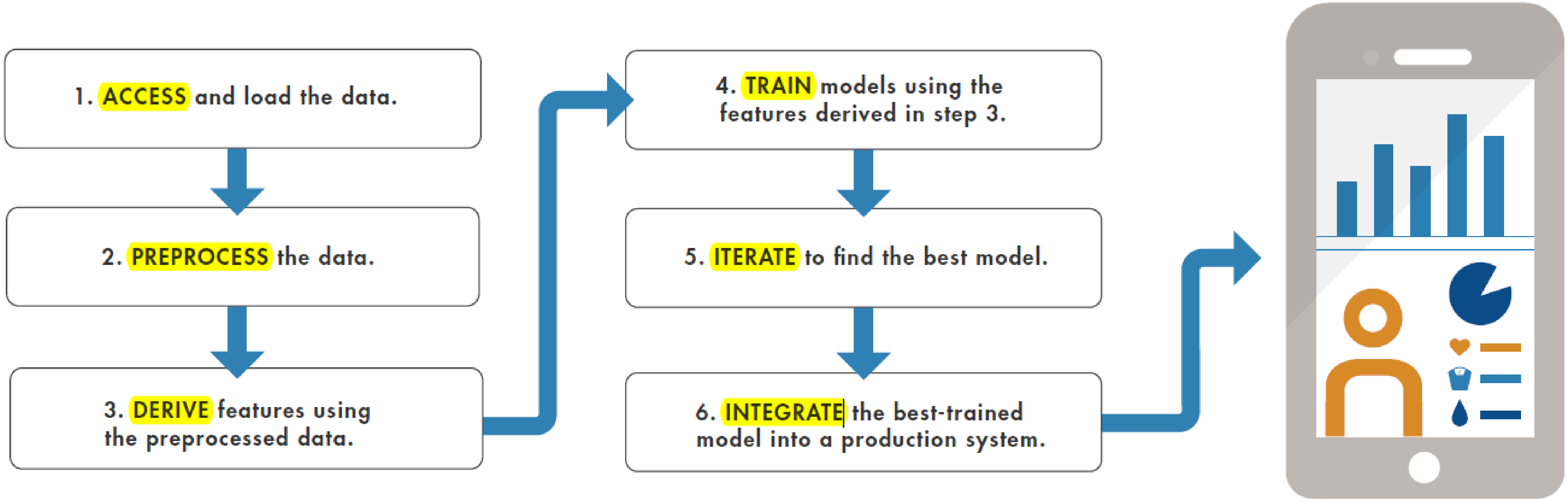
Machine Learning Workflow

A step-by-step example of the MATLAB ML application

Workflow of ML



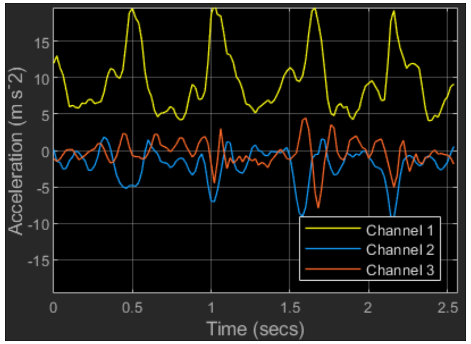
Workflow of ML in MALTAB

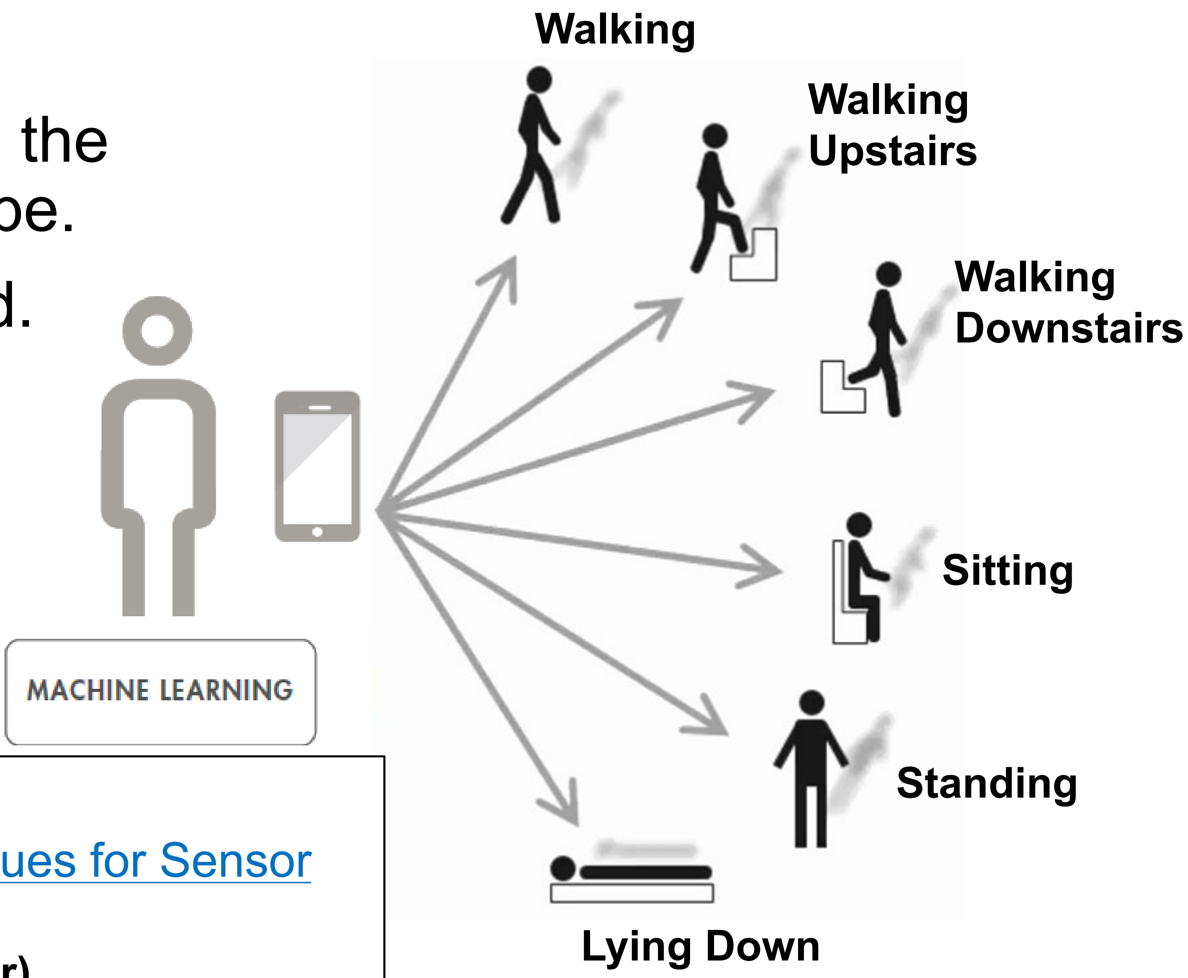


Machine Learning with MATLAB, Section 2

Classifying Physical Activities

- A cell phone health-monitoring app.
- **Inputs:** three-axial sensor data from the phone's accelerometer and gyroscope.
- **Responses:** the activities performed.

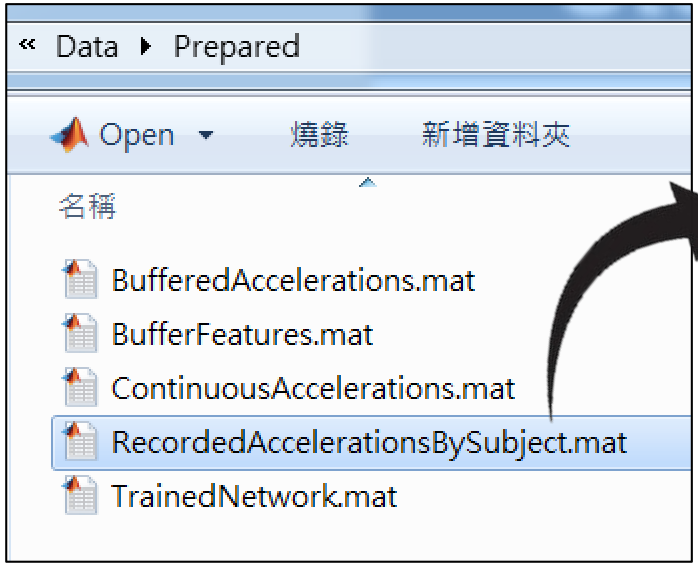




Gabriele Bunkheila, MathWorks
Signal Processing and Machine Learning Techniques for Sensor Data Analytics
- **Download Example Data** (for R2019b or MAC user)

Step 1: Load/Acquire the Data

>> DataPreparation

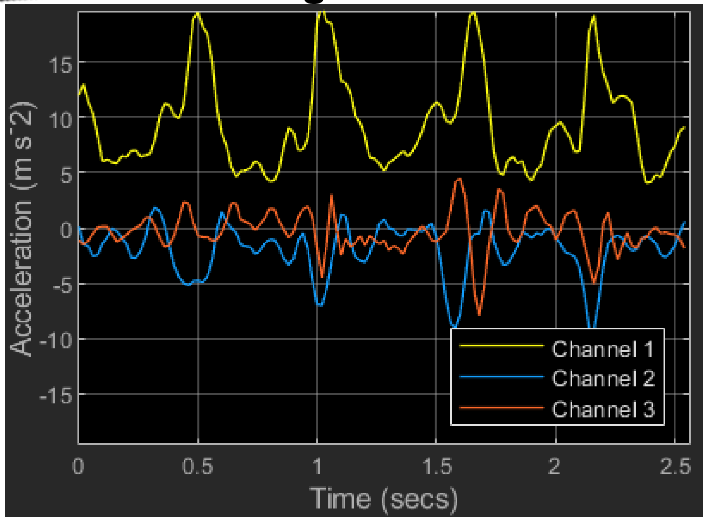


1x30 struct with 2 fields

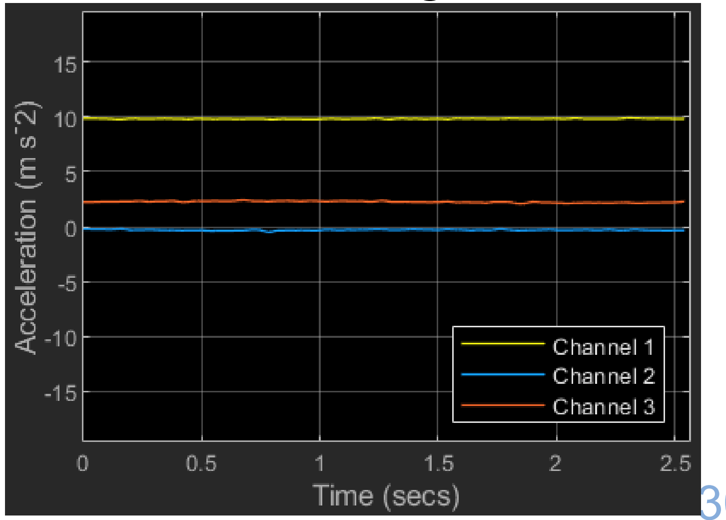
Fields	actid	totalacc
1	*22272x1 double*	*22272x3 double*
2	*19392x1 double*	*19392x3 double*
3	*21888x1 double*	*21888x3 double*
4	*20352x1 double*	*20352x3 double*
5	*19392x1 double*	*19392x3 double*
6	*20864x1 double*	*20864x3 double*
7	*19776x1 double*	*19776x3 double*
8	*18048x1 double*	*18048x3 double*
9	*18496x1 double*	*18496x3 double*
10	*18880x1 double*	*18880x3 double*
11	*20288x1 double*	*20288x3 double*
12	*20544x1 double*	*20544x3 double*
13	*20992x1 double*	*20992x3 double*

actid
1: Walking
2: WalkingUpstairs
3: WalkingDownstairs
4: Sitting
5: Standing
6: Lying Down

Walking Downstairs



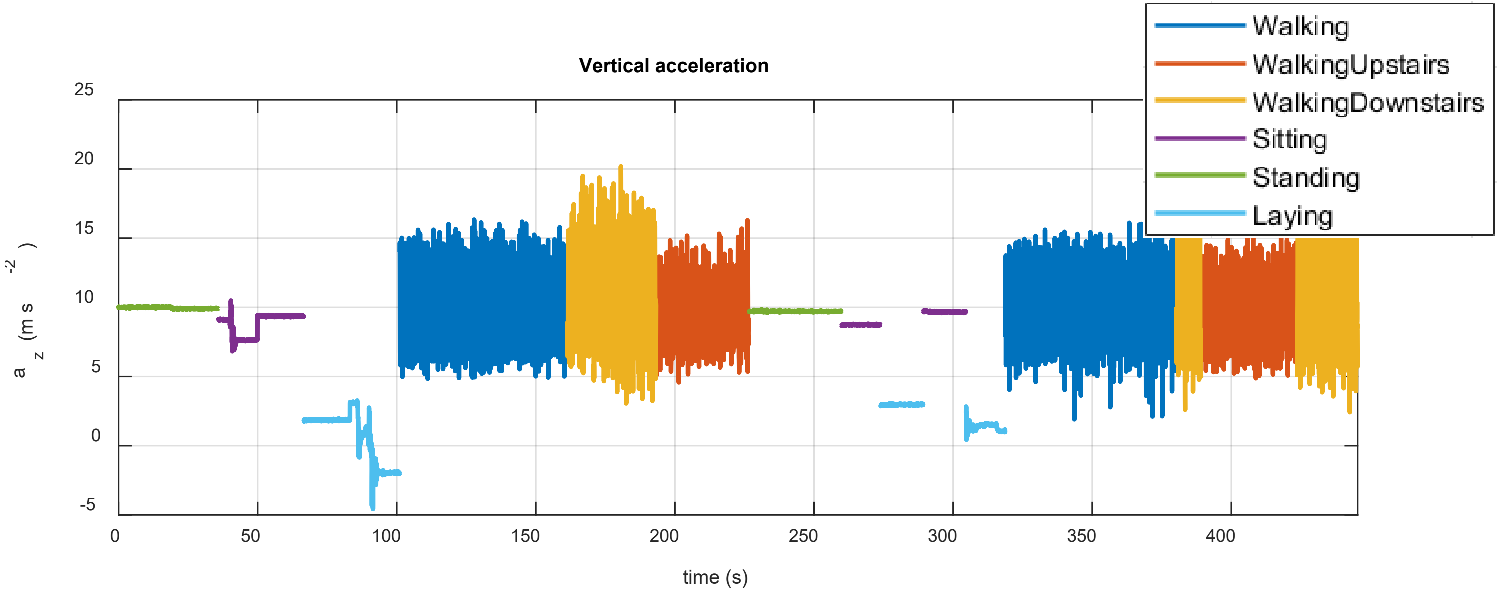
Standing

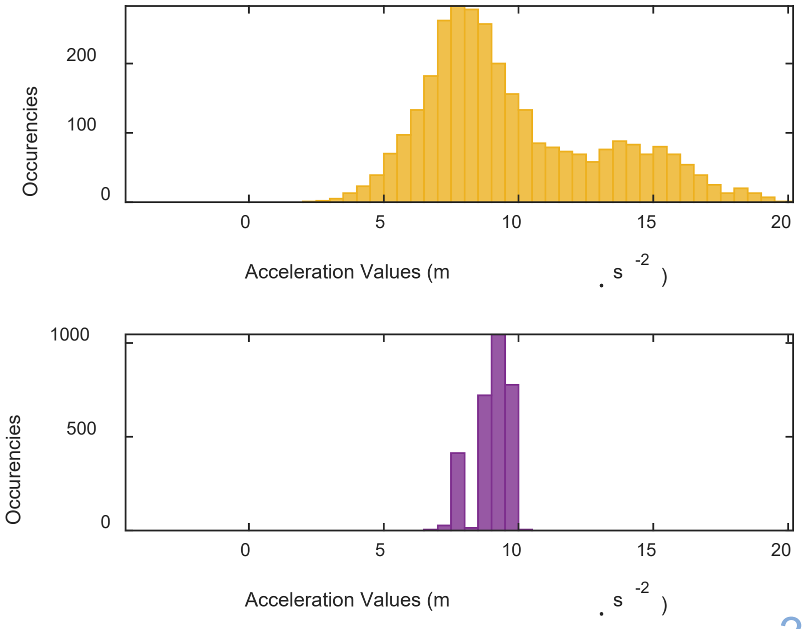


Step 2: Preprocess the Data (1/2)

- **Look for outliers**–data points that lie outside the rest of the data.
  - **[Caution]** Decide whether the outliers can be ignored or whether they indicate a phenomenon that the model should account for.
- **Check for missing values** (data lost due to the connection dropped during recording).

>> isoutlier





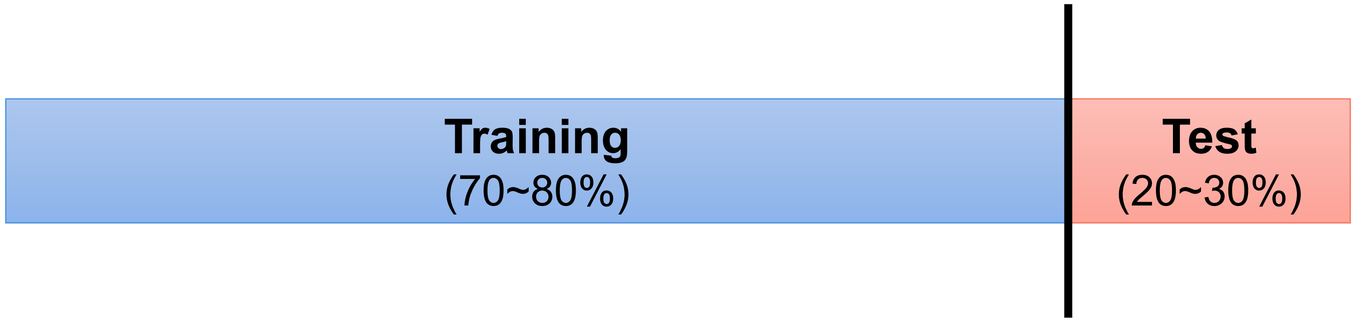
Step 2: Preprocess the Data (2/2)

- Remove gravitational effects from the accelerometer data
  - To focus on the movement of the subject, not the movement of the phone.

>> filter

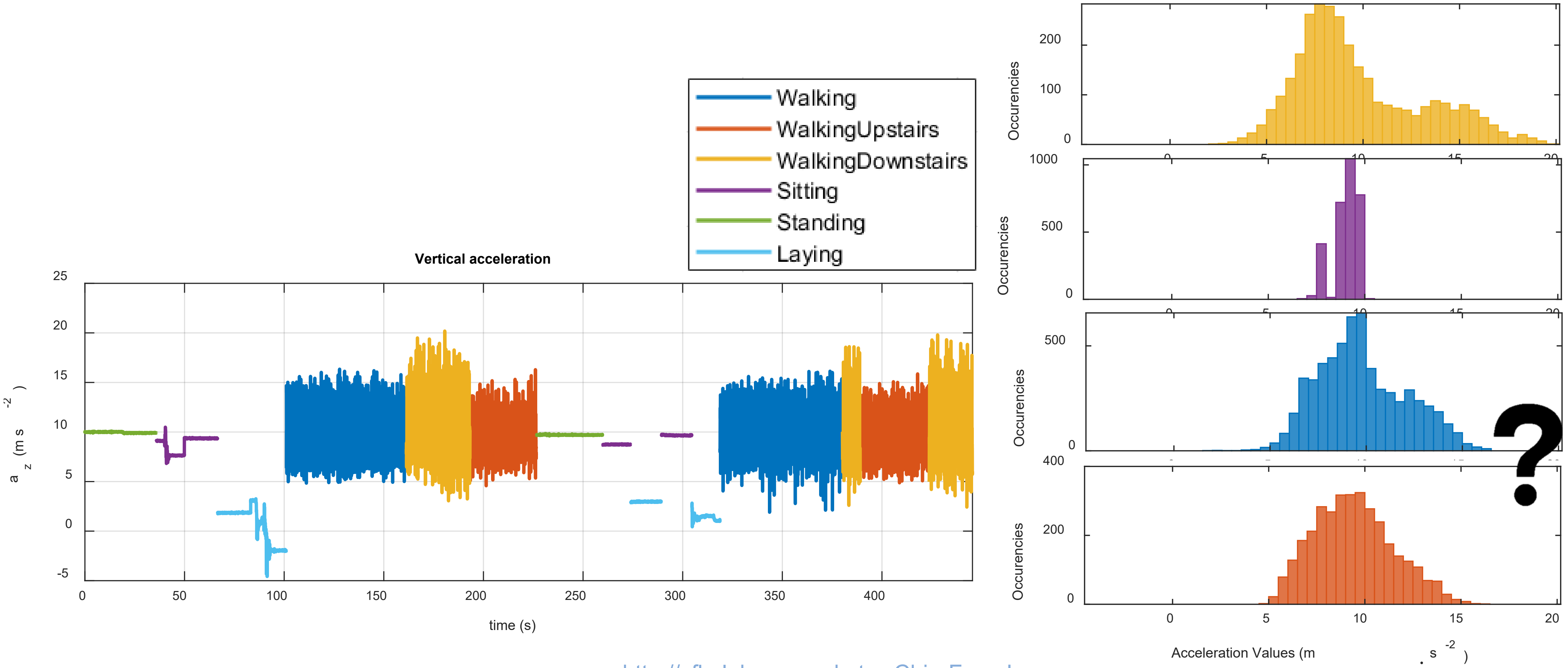
- Divide the data into two sets
  - **Training set:** to build models
  - **Test set:** to assess the model performance

>> cvpartition



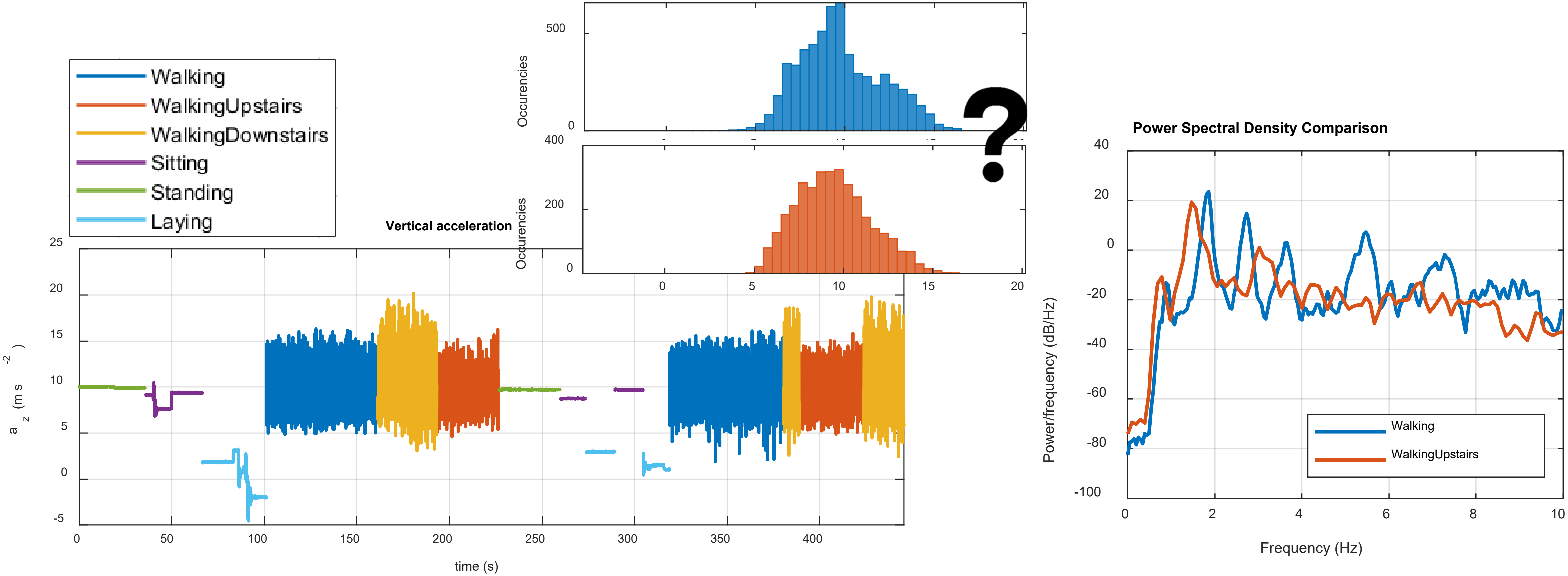
Step 3: Derive Features (1/3)

- Amplitude-only methods are often not enough!



Step 3: Derive Features (2/3)

- Add in the frequency features of the accelerometer data.



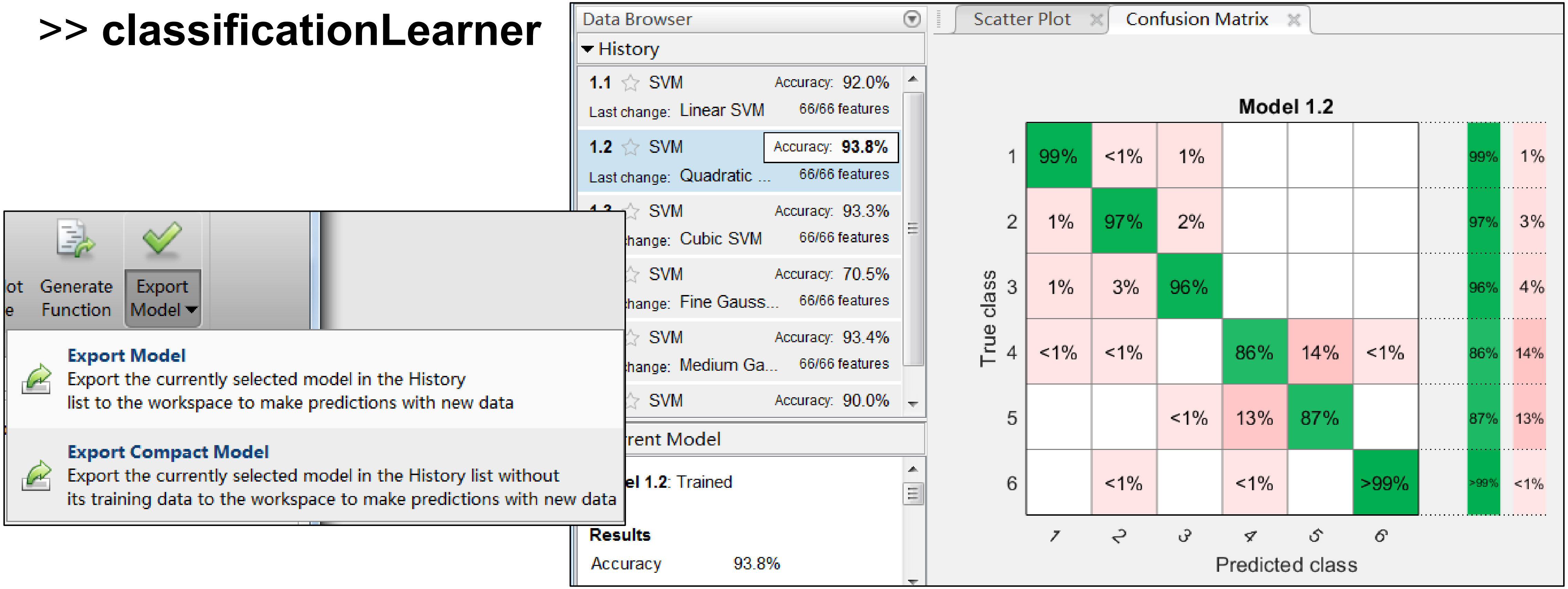
Step 3: Derive Features (3/3)

Some examples of useful features…

Data Type	Feature Selection Task	Techniques
Sensor data	Extract signal properties from raw sensor data to create higher-level information	**Peak analysis** – perform an fft and identify dominant frequencies **Pulse and transition metrics** – derive signal characteristics such as rise time, fall time, and settling time **Spectral measurements** – plot signal power, bandwidth, mean frequency, and median frequency
Image and video data	Extract features such as edge locations, resolution, and color	**Bag of visual words** – create a histogram of local image features, such as edges, corners, and blobs **Histogram of oriented gradients (HOG)** – create a histogram of local gradient directions **Minimum eigenvalue algorithm** – detect corner locations in images **Edge detection** – identify points where the degree of brightness changes sharply

Step 4: Build and Train the Model

```
>> isAllPreparedDataAvailable
>> load('BufferFeatures.mat')
>> featTable = featuresTable(feat, featlabels, actid);
>> classificationLearner
```

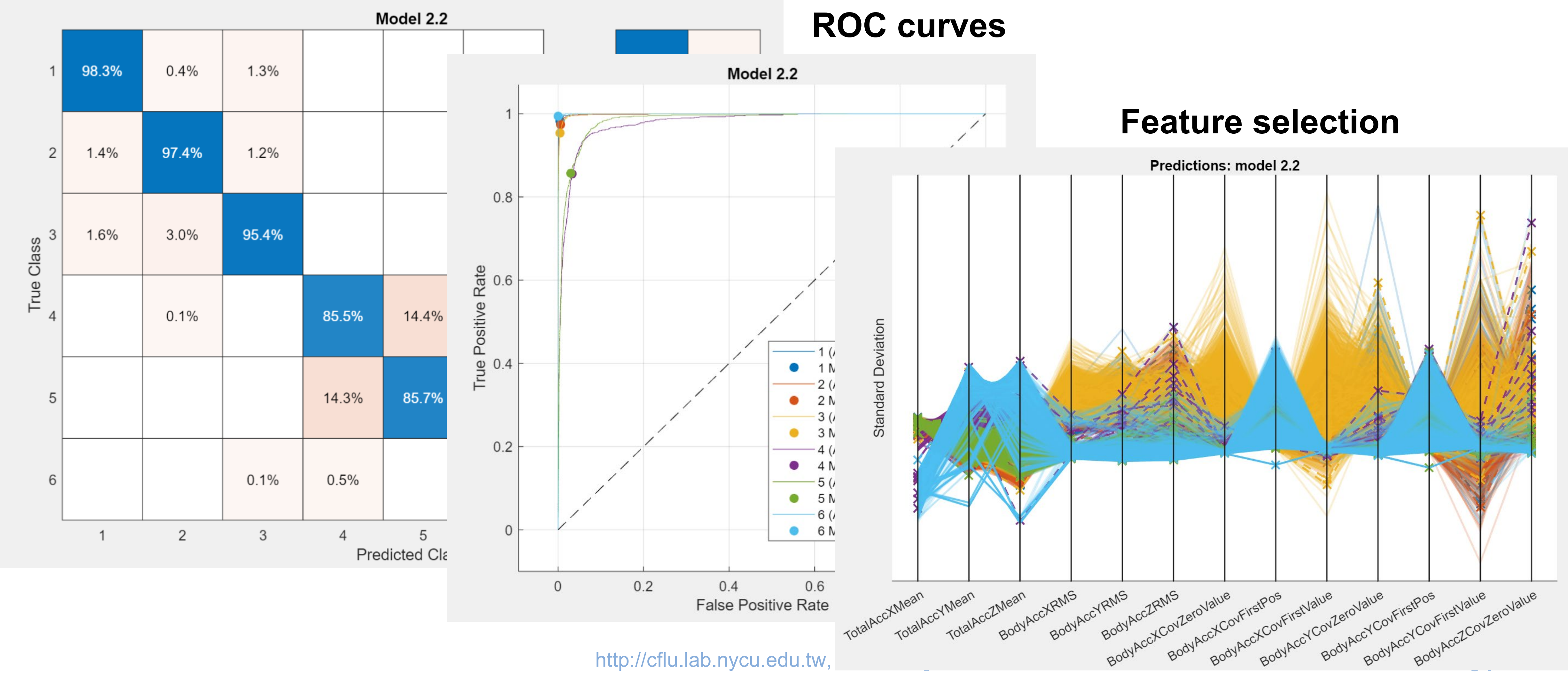


Model Performance

Confusion matrix

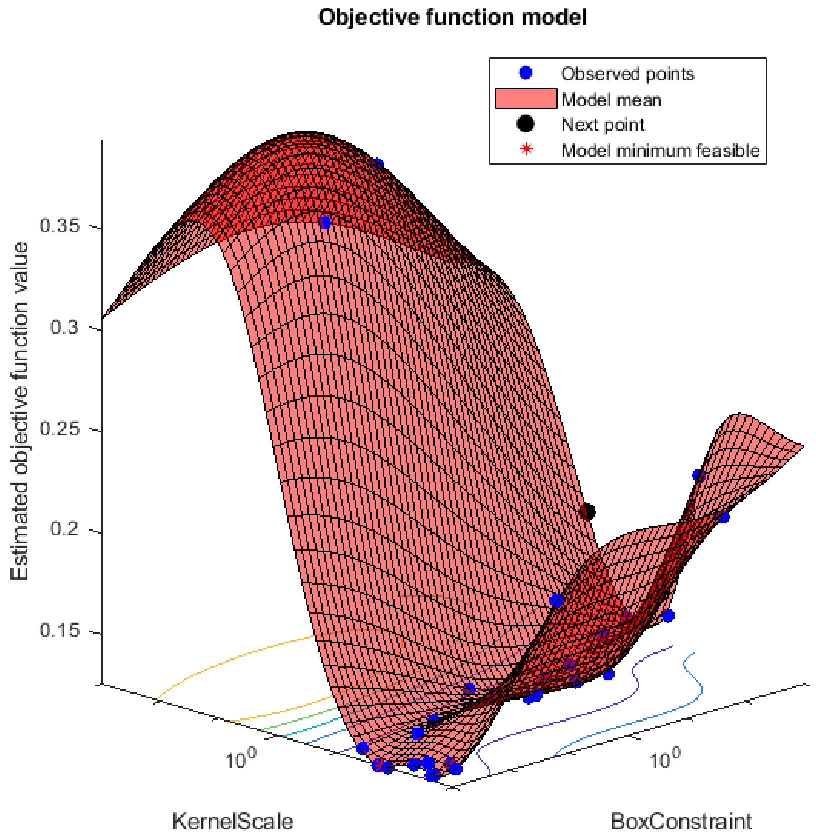
ROC curves

Feature selection



Step 5: Improve the Model

- **Tune/Optimize the hyperparameters**
- **Perform feature reduction**
  - Correlation analysis between features
  - Principal component analysis (PCA)
- **Perform feature selection**
  - Sequential feature reduction
  - Minimum Redundancy Maximum Relevance (mRMR) algorithm
  - least absolute shrinkage and selection operator (LASSO) algorithm
- **Reduce the model complexity**
  - Pruning branches from a decision tree
  - Removing learners from an ensemble



THE END

Contact:
盧家鋒 alvin4016@nycu.edu.tw