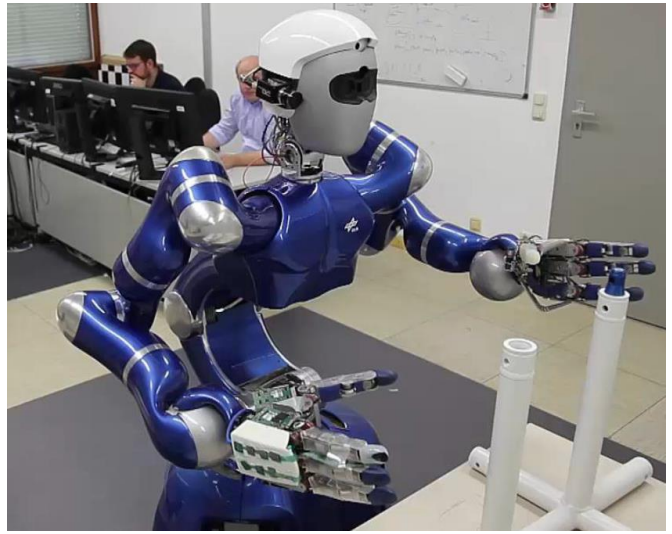
MATLAB EXPO 2018

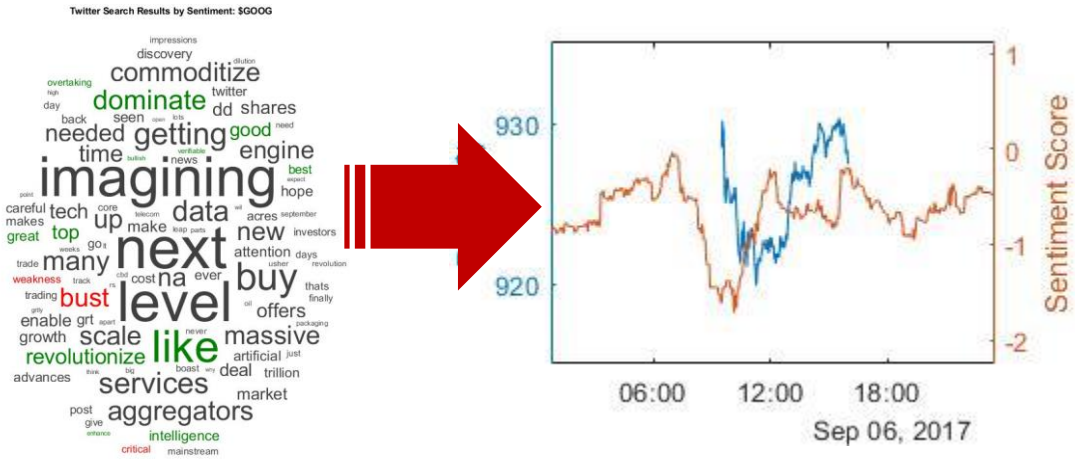
Unlocking the Power of Machine Learning

Rainer Muemmler, MathWorks

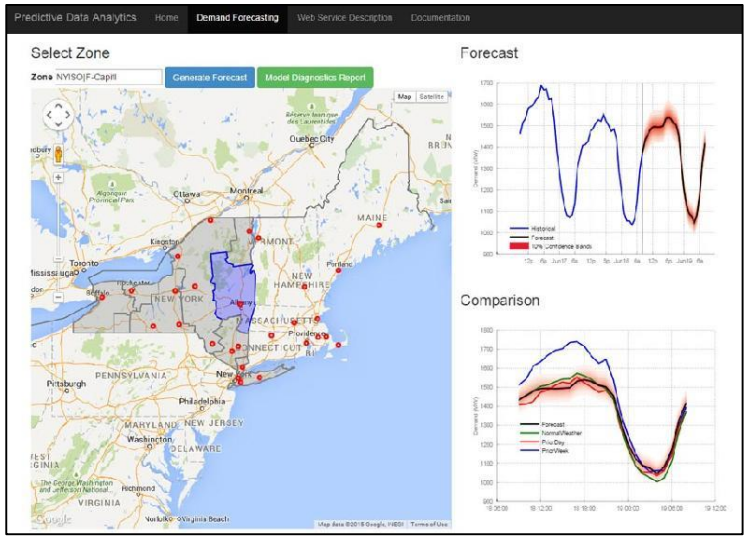
Machine Learning has driven Innovation



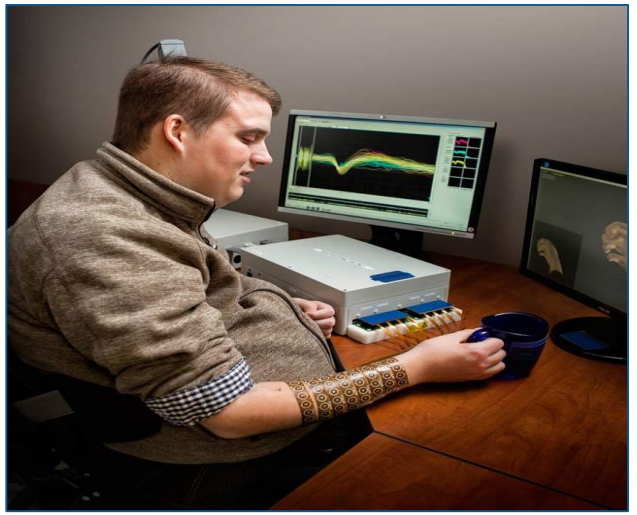
Robots mimic complex human behaviors



Sentiment Analysis in Finance



Electric Grid Load Forecasting



Restore Arm Control for Quadriplegic

Outline

Machine Learning workflow and its challenges

Overview of Types of Machine Learning

Developing a Heart Sound Classifier

Applying Deep Learning

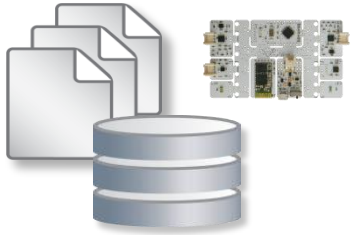
Key takeaways

- Cover complete workflow (exploration to deployment)
- Make machine learning easy
- Support for Deep Learning

Challenges in Developing Machine Learning Applications

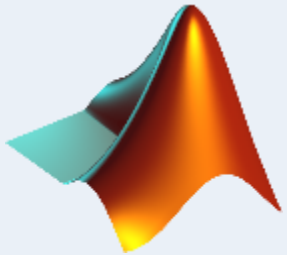
1. Access Data

Sensors
Various Protocols



2. Explore and Pre-Process

Diverse data
Clean messy data
Discover patterns



3. Feature Extraction

Domain Knowledge
Select Features

4. Build Models

Many Algorithms
Tune Parameters

5. Deploy

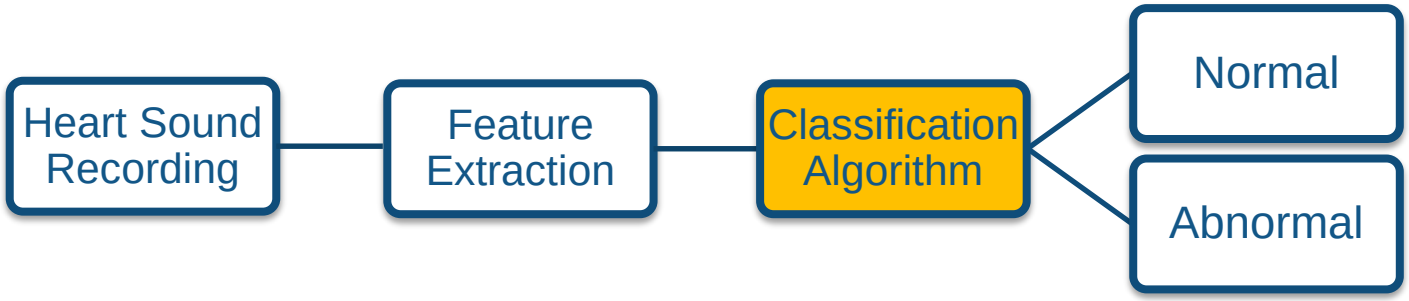
Different platforms
Size/Speed

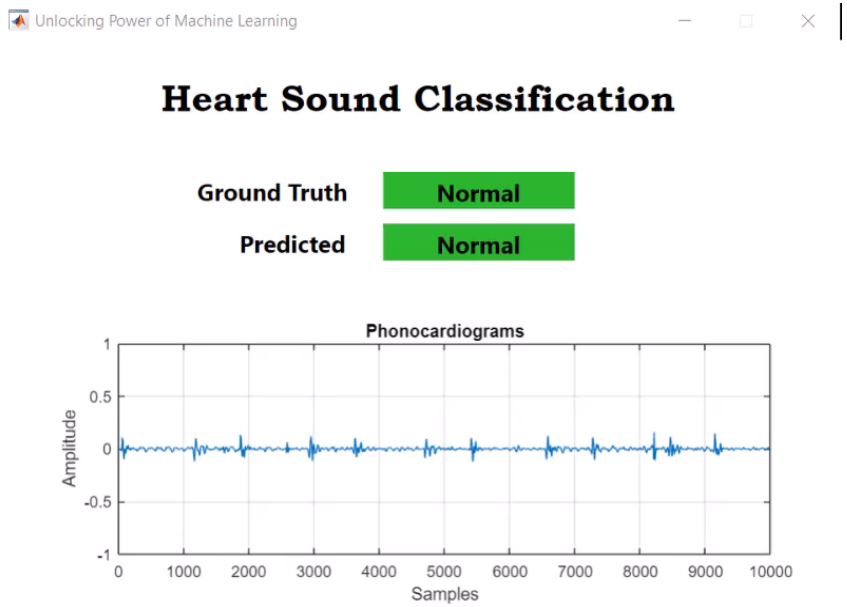
Case Study: Heart Sound Classifier

Motivation

- Heart sounds require trained clinicians for diagnosis
- Lowered FDA requirements renewed interest

Goal: build a classifier and deploy in portable device

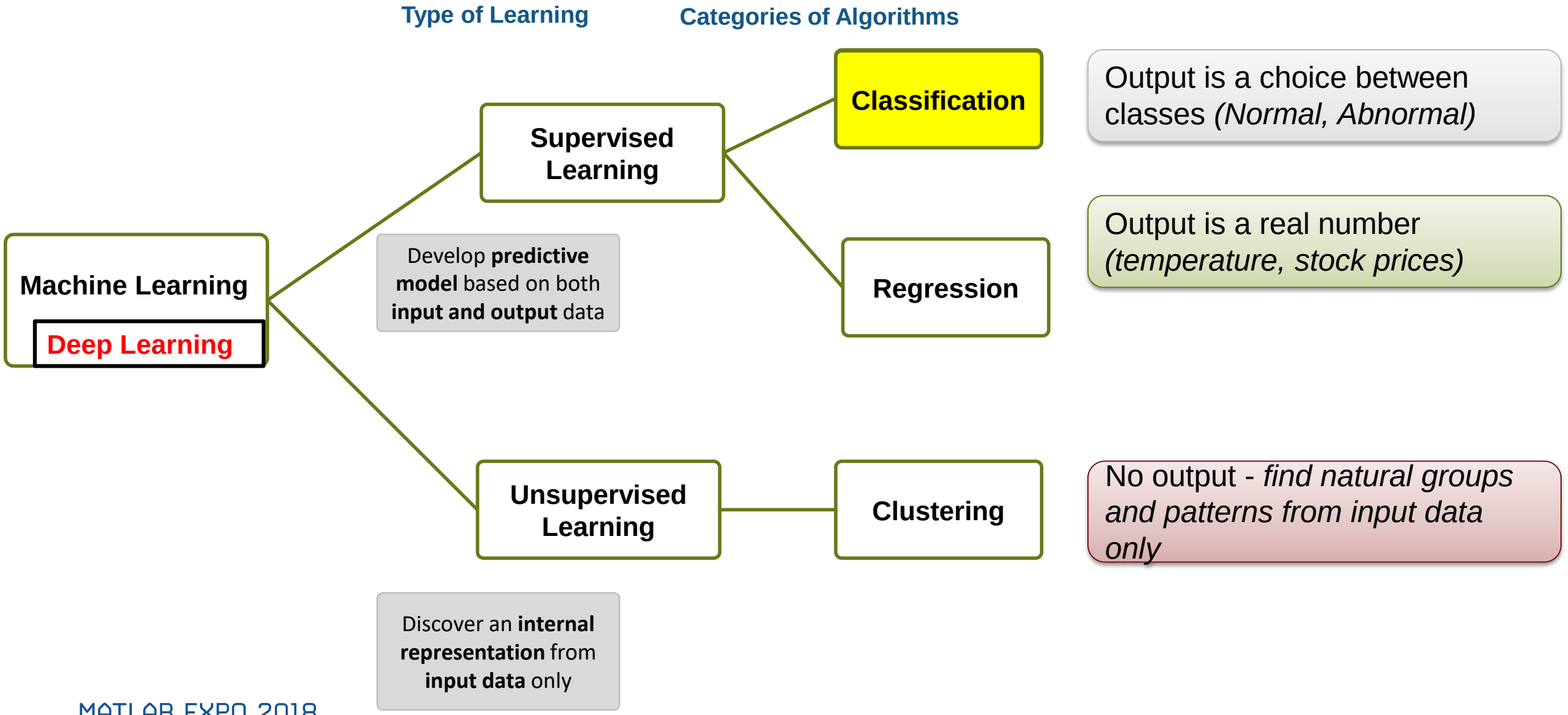




Data: Heart sound recordings (phonocardiogram):

- From PhysioNet Challenge 2016
- 5 to 120 seconds long audio recordings

Different Types of Learning



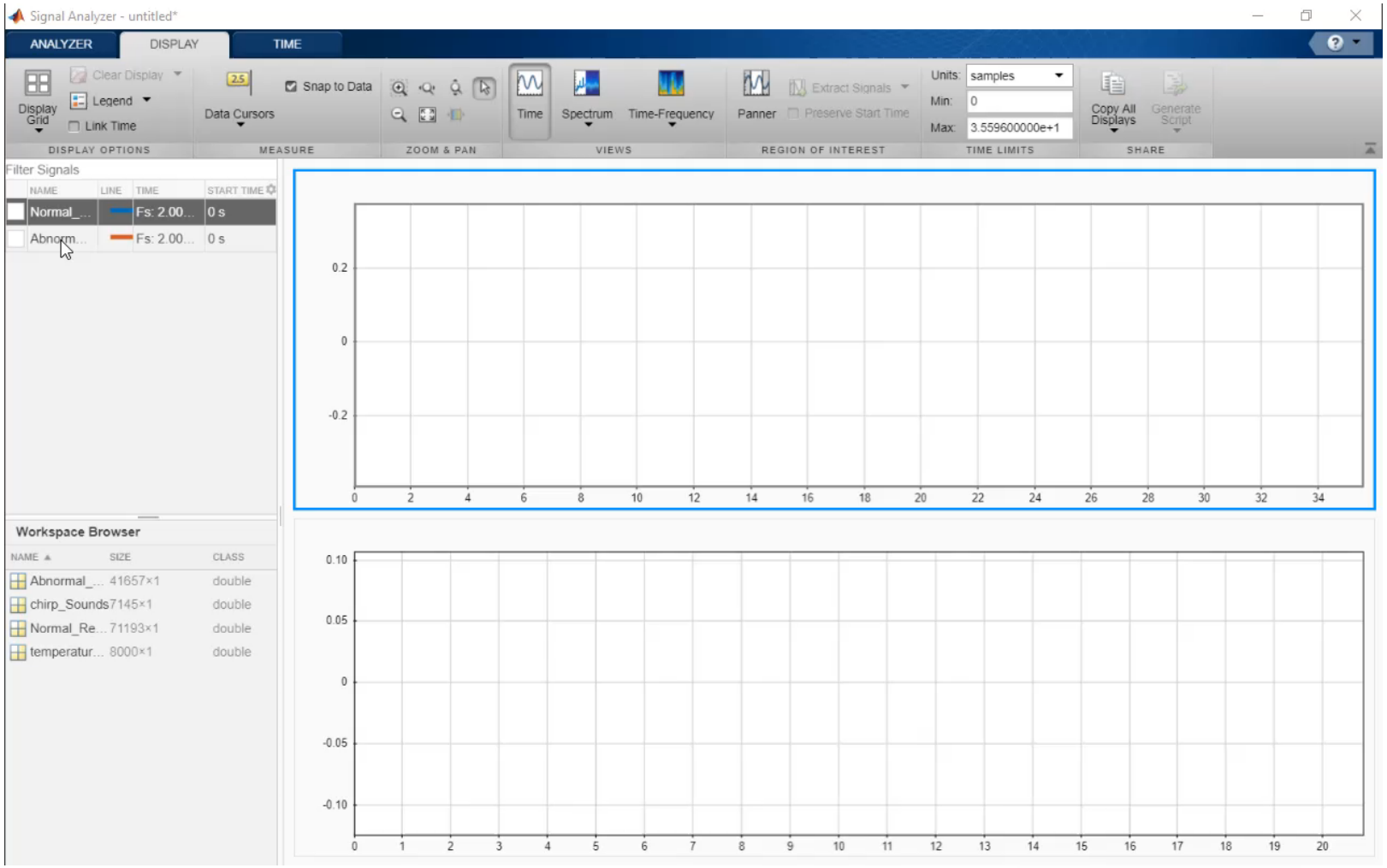
Step 1: Access & Explore Data

Challenges:

- Different sampling rates
- Signal Management
- Large datasets (“big data”)

Easy Exploration of Data

- Time domain
- Frequency domain
- Time-Frequency domain



Signal Analyzer: Visual Data Exploration

R2018a

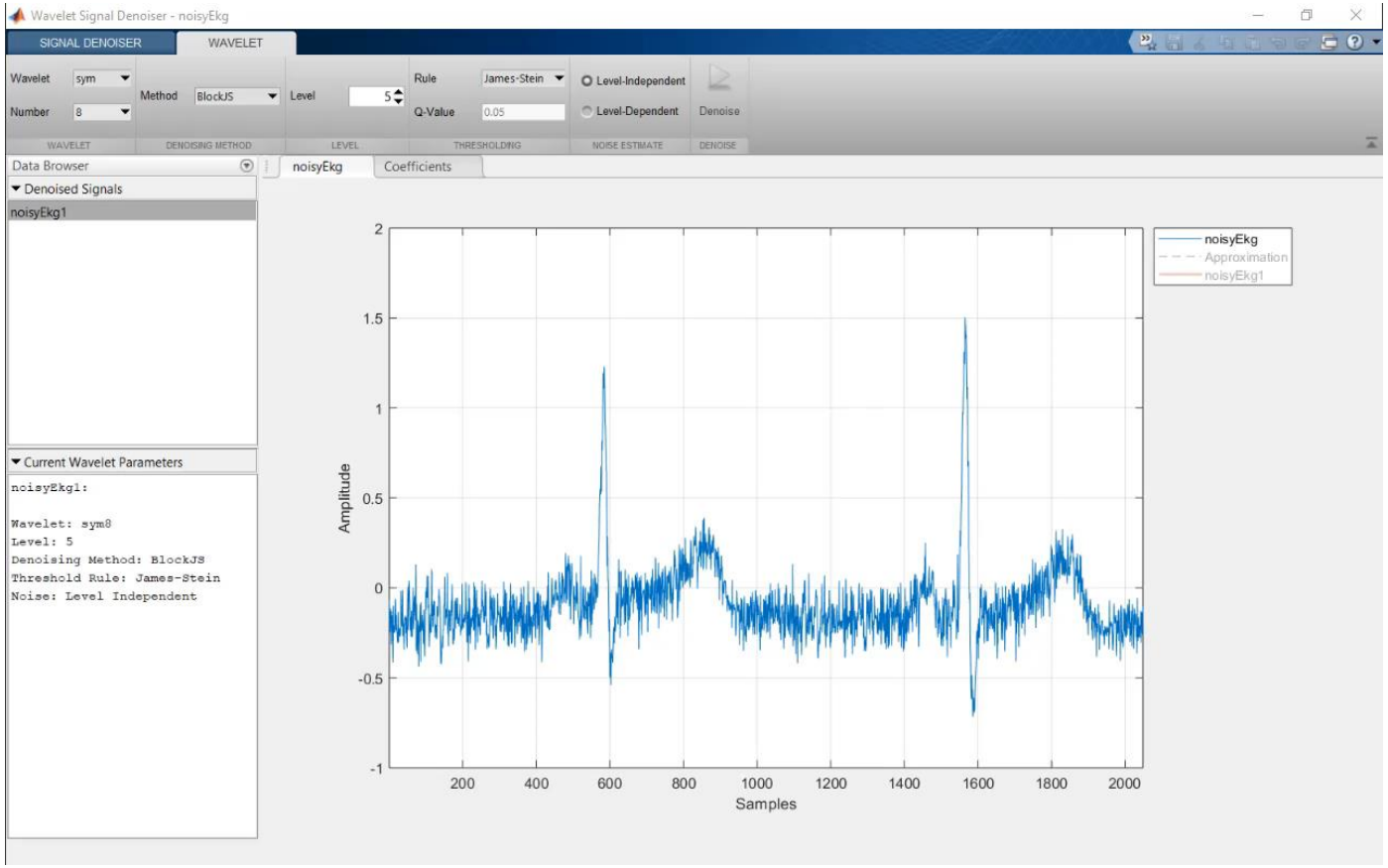
Step 2: Pre-process Signals

Challenges

- Preserving sharp features
- Overlap of signal and noise spectra

Automatic Denoising

Generate MATLAB code



Signal Pre-processing without writing any code

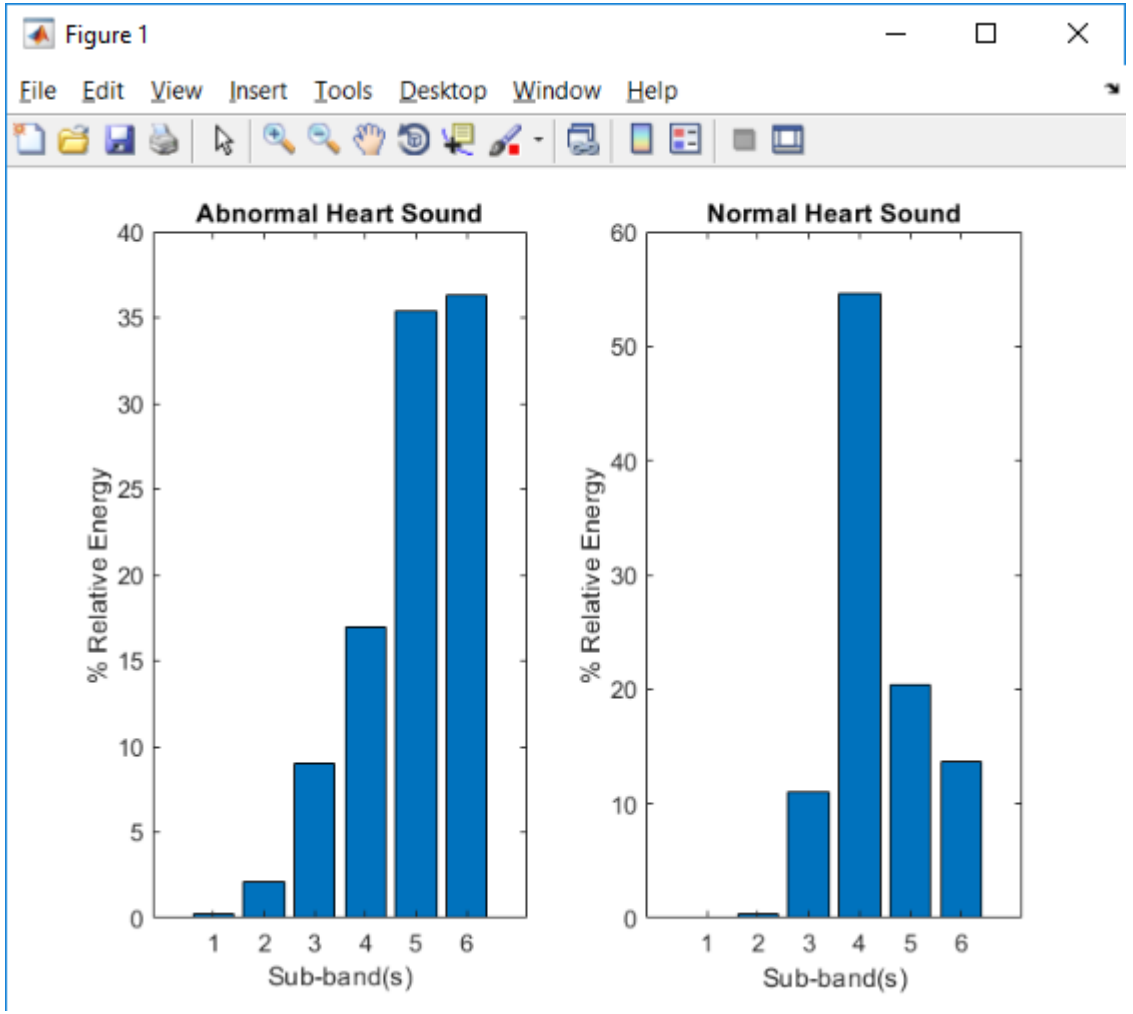
Step 3: Extract Features

Challenges

- Find features for non-stationary signals
- Features occurring at different scales
- Feature selection

Spectral features:

- Mel-Frequency Cepstral Coefficients
- Octave band decomposition with Wavelets



Step 4: Train Models

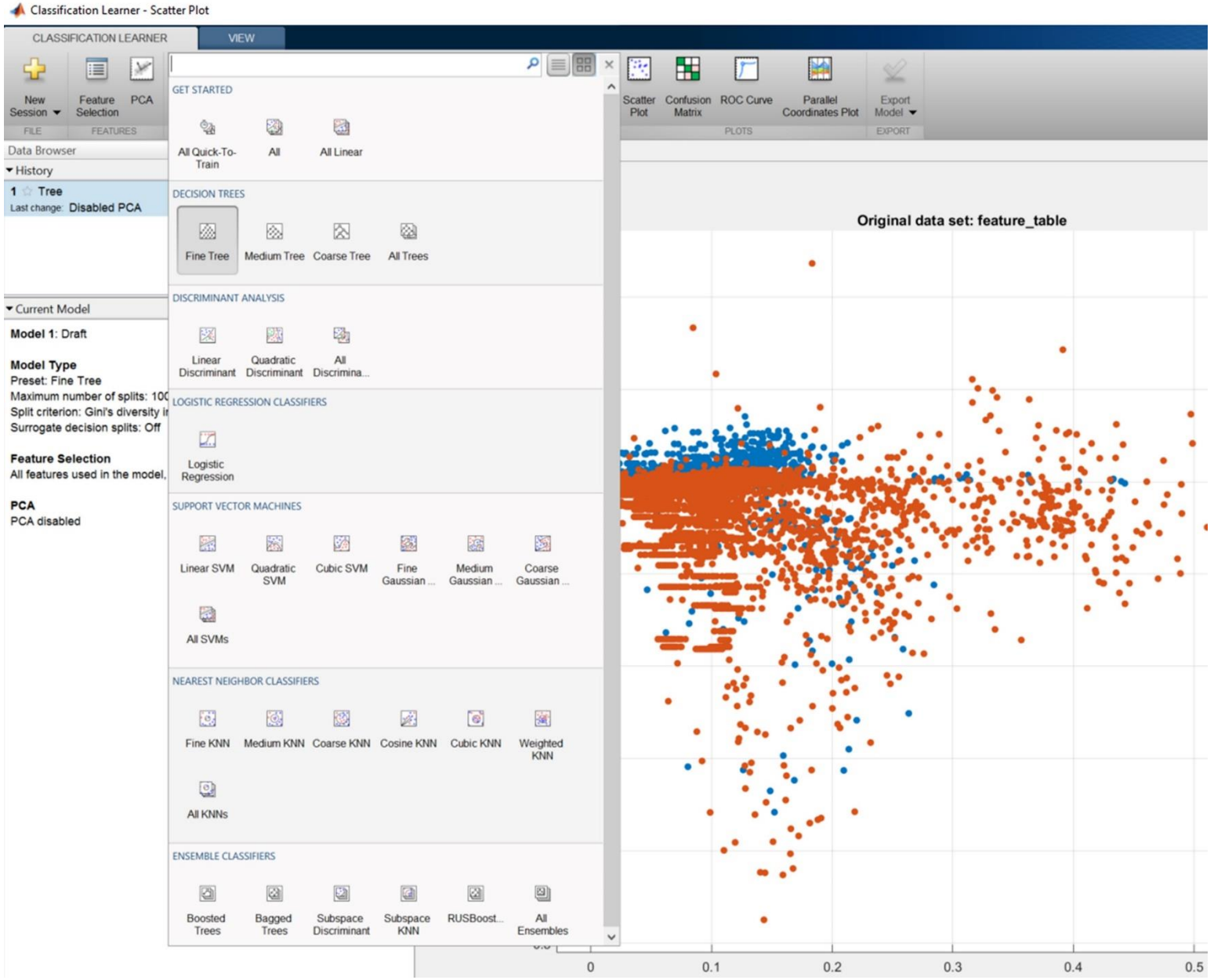
Challenges:

- Knowledge of machine learning algorithms
- Scale to large data sets

Quickly train model in App

- Define crossvalidation
- Try all popular algorithms
- Analyze performance: 93% on test data

Scale to large data sets without recoding: “Tall” arrays



Model Training with Classification Learner

Step 4 Cont'd: Optimize Model

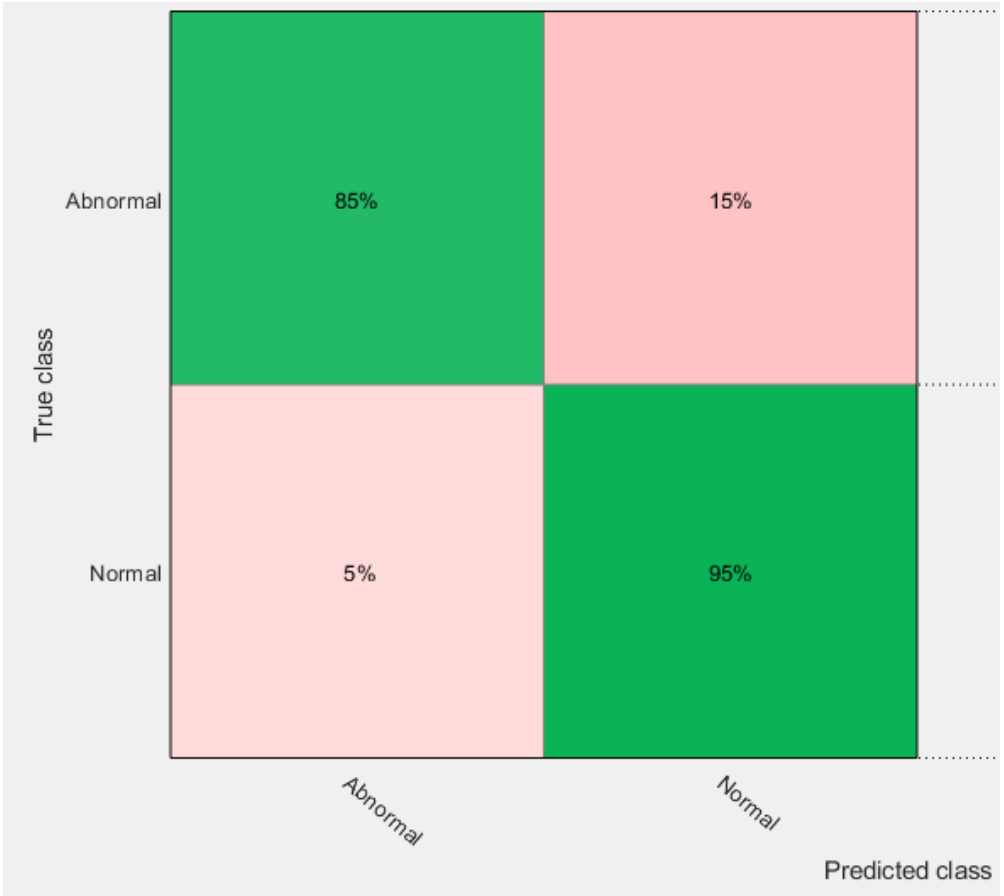
Challenges:

- Manual parameter tuning tedious
- Identify additional improvements

Iterative Model Optimization

- Bayesian Optimization of parameters
- Visually analyze performance
- Adjust for imbalances (data or severity of misclassifications)

Class	Distribution
Normal	75%
Abnormal	25%



Step 5: Deploy

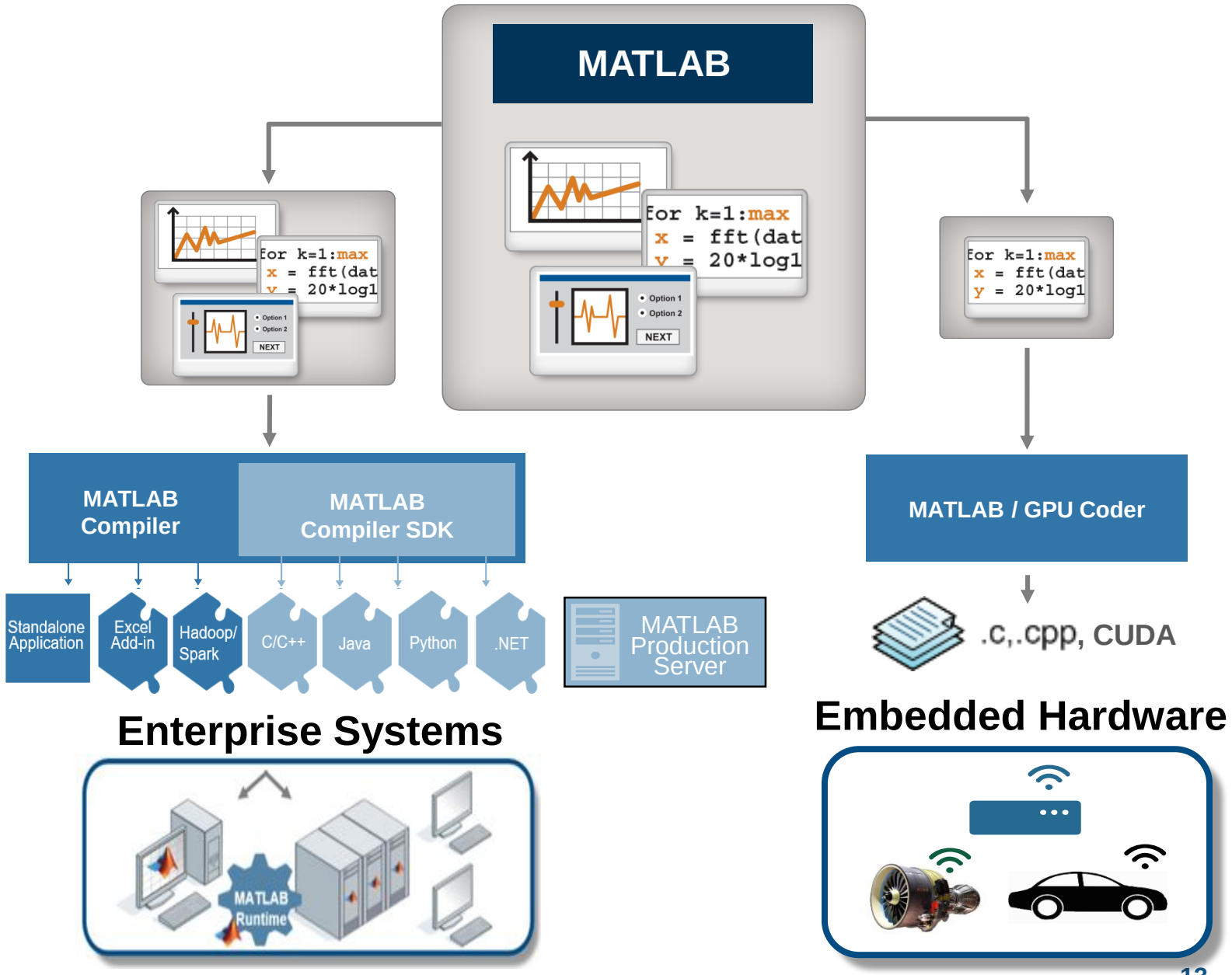
Challenges:

- Different target platforms
- Hardware requirements (Size, Speed, Fixed point, etc)

Deployment options:

- Generate Code (C, HDL, PLC) for Embedded System
- Compile MATLAB, scale using MPS for Enterprise systems

Apply automated feature selection to reduce model size

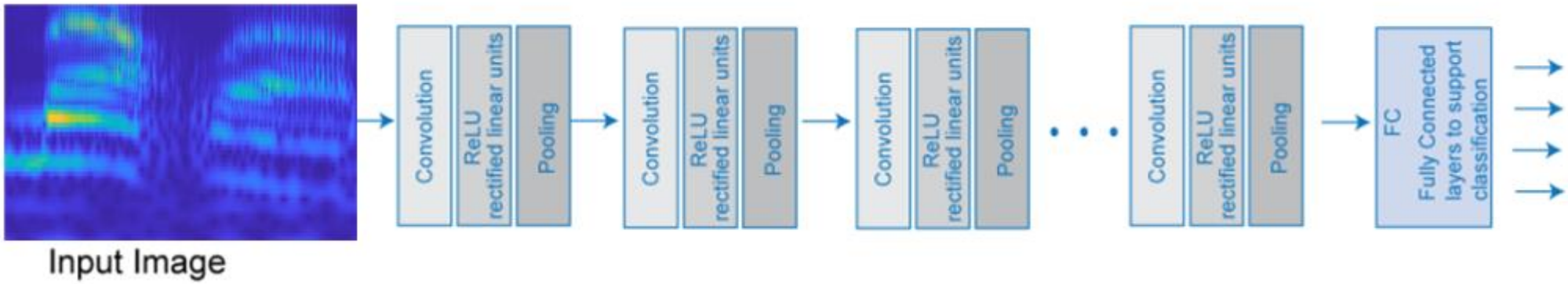


Deep Learning on Signals

Supervised Classification using Neural Nets with many layers

1. Convolutional Neural Networks (CNN)

- A versatile and flexible approach for Deep Learning
- Apply to signals by converting to time-frequency representation:



2. Long short-term memory networks (LSTM)

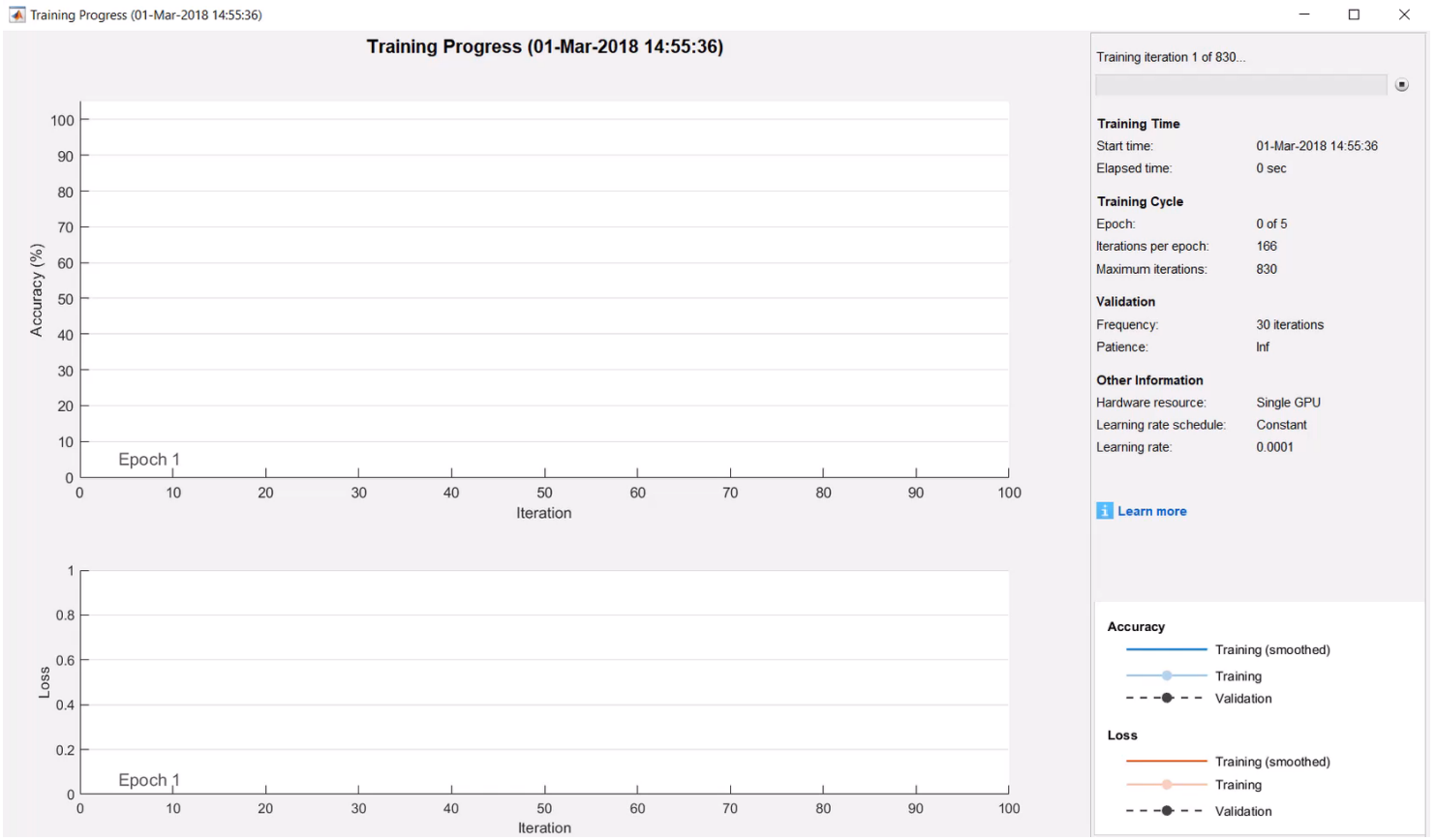
Apply Deep Learning to Heart Sound Classifier

Steps

- Signal ➔ Time-Frequency
- Continuous Wavelet Transform
- Transfer Learning with GoogleNet

Results

- Achieves 90% accuracy
- Just 10 lines of code



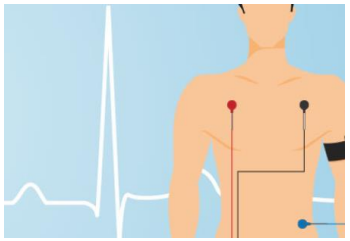
Deep Learning Training

R2017b

Recap: Making Machine Learning Easier

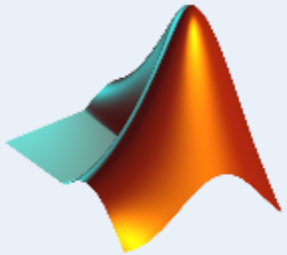
1. Access Data

Support for industrial sensors, phones, etc.



2. Explore and Pre-Process

Visual Exploration



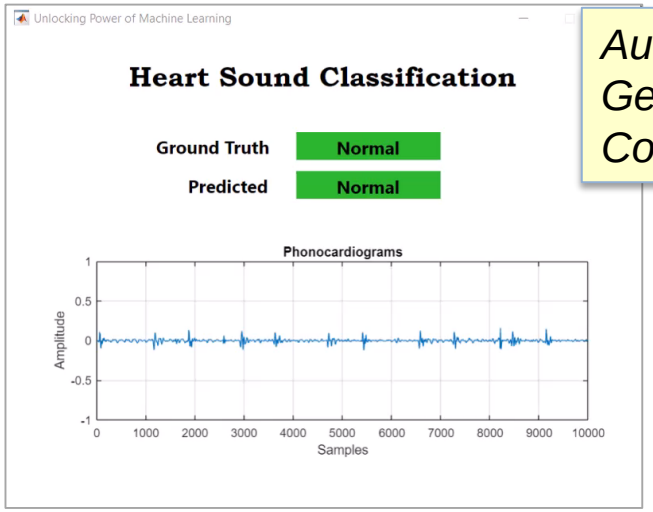
3. Feature Extraction

Wavelets
Feature Selection

4. Build Models

Quickly compare models in App
Automatically tune parameters
Explore Deep Learning

5. Deploy



Automatically Generate C/CUDA Code

Key takeaways

Empower engineers to be productive in data science!

- Cover complete workflow (exploration to deployment)
- Make machine learning easy
- Support for Deep Learning

Learn More

Complete user story for Battelle's "NeuroLife" system

Download Heart Sounds Classification application from File Exchange

Watch "Machine Learning Using Heart Sound Classification"

Read:

- Machine Learning with MATLAB
- What is Deep Learning?

Battelle Neural Bypass Technology Restores Movement to a Paralyzed Man's Arm and Hand

Challenge

Restore arm and hand control to a quadriplegic man by processing signals from an electrode array implanted in his brain

Products Used

- MATLAB + Wavelet Toolbox

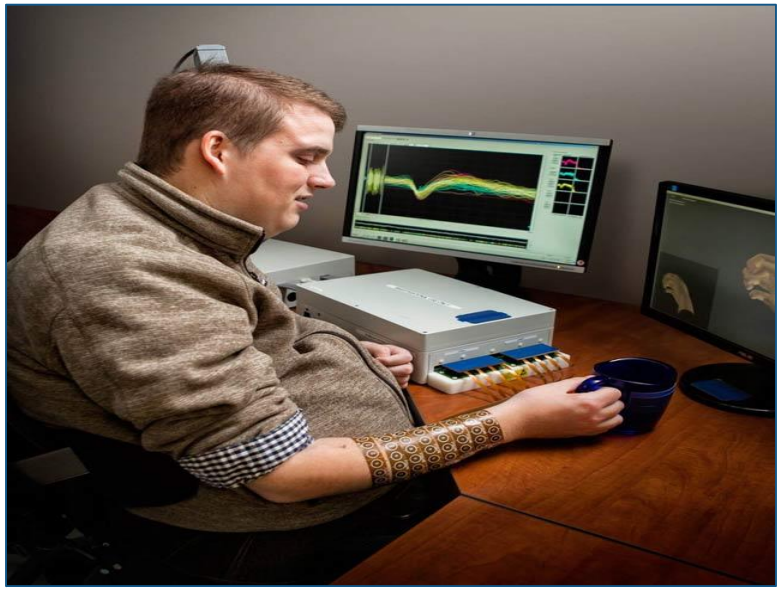
Approach

- Used MATLAB to analyze signal samples
- Generated compact feature vectors using wavelets **– wavelets allowed the researchers to extract the important information from the signals for classification**.
- Applied machine learning to classify patterns mapped to movements, and
- Generate actuation signals for a neuromuscular electrical stimulator

Results

- Control over paralyzed hand and arm restored
- Real-time processing performance achieved

Link to full user story



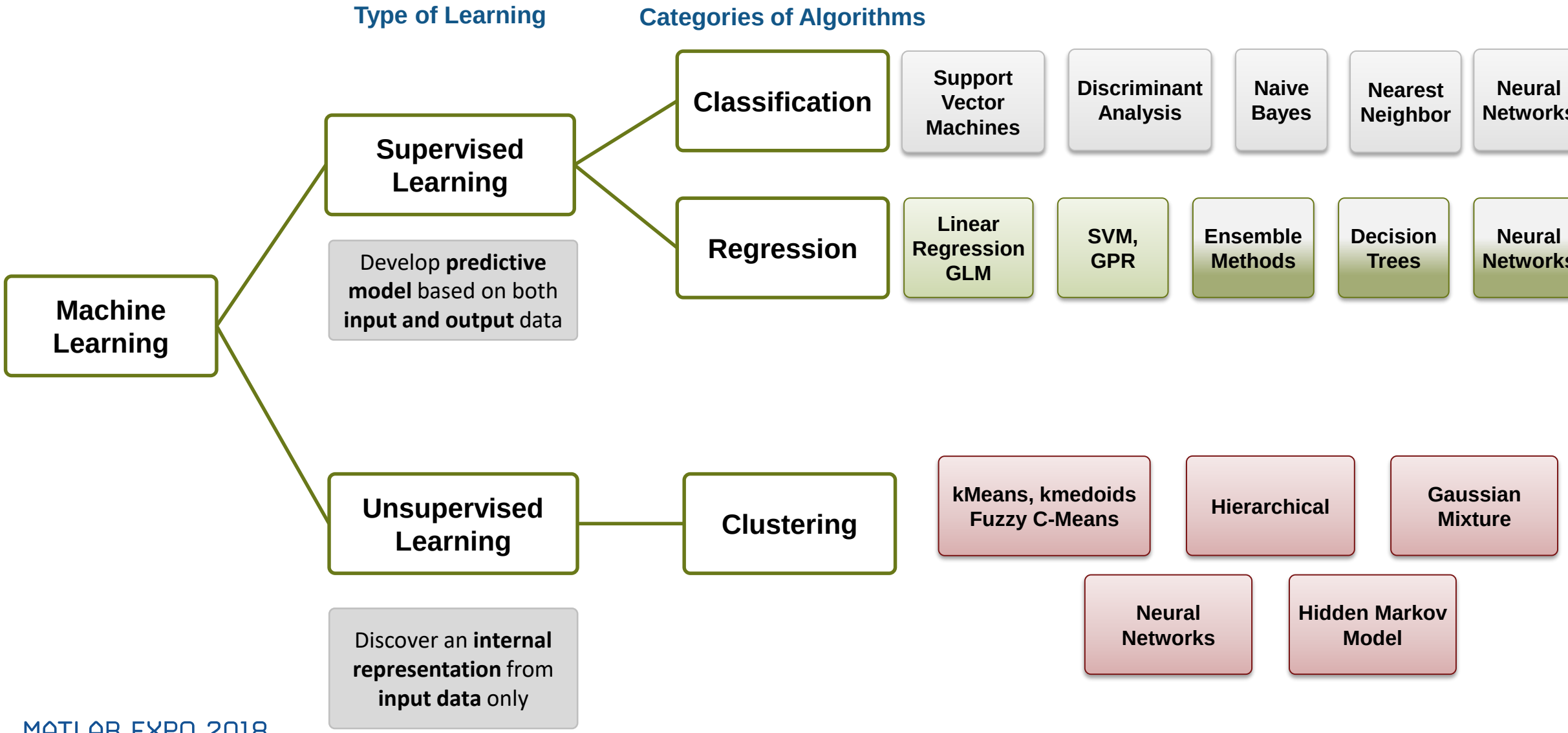
Patient using the Battelle NeuroLife system.

"The algorithms we developed using MATLAB gave the participant back basic control of his arm and hand. By the end of the study, he could grip a bottle, pour out its contents, and set it down, as well as pick up a stir stick and execute a stirring motion."

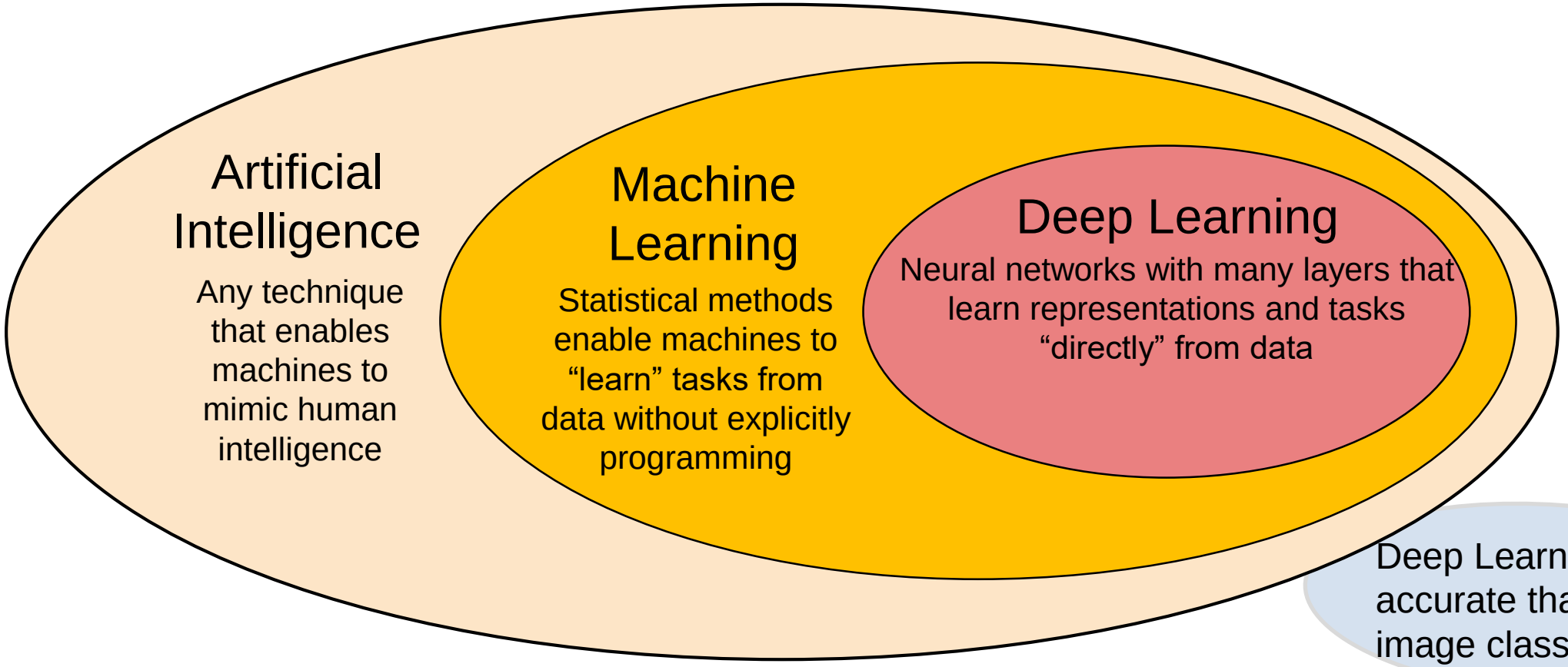
David Friedenberg

Battelle

Different Types of Learning

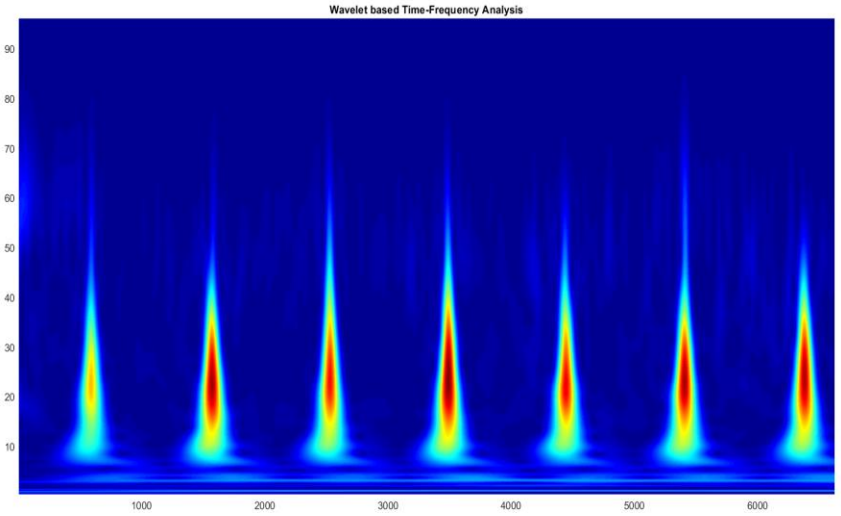


AI, Machine Learning, and Deep Learning



How to generate Time-Frequency Representation for Signals ?

- Wavelets can localize spectral content as a function of time
- Enhance subtle information in signals using wavelet time-frequency approach
- Feed time-frequency representations of your signals
- Generate time-frequency representations only with **two** lines of MATLAB code



MATLAB “works with” Open Source

3 Ways to Interoperate:

More productive where we differentiate (7 Truths!)

Enable Collaboration where we complement

- Exchange work among teams
- Choice of tool