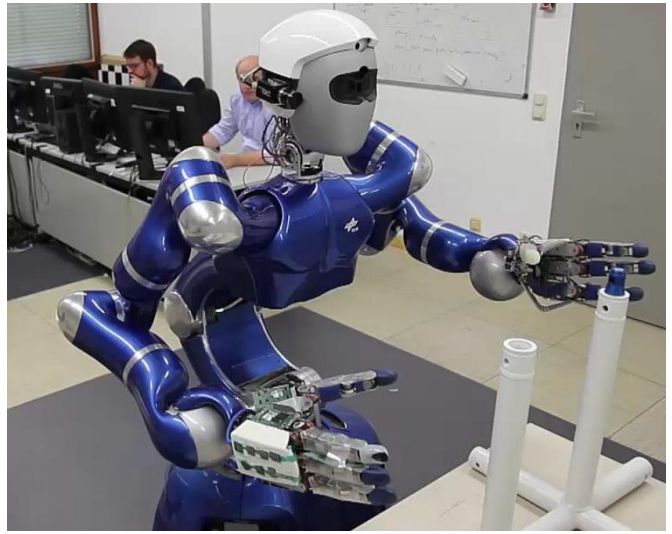
# MATLAB EXPO 2018

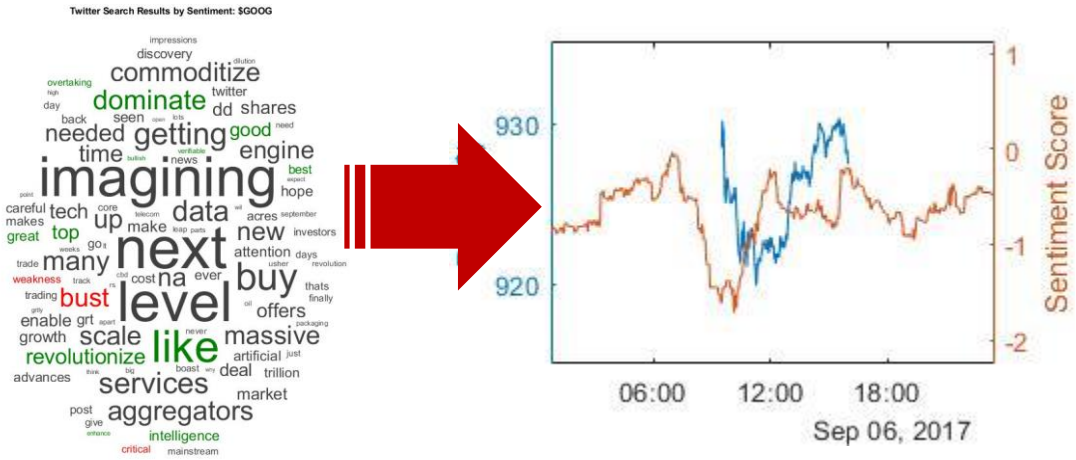
## 释放机器学习的力量

阮卡佳

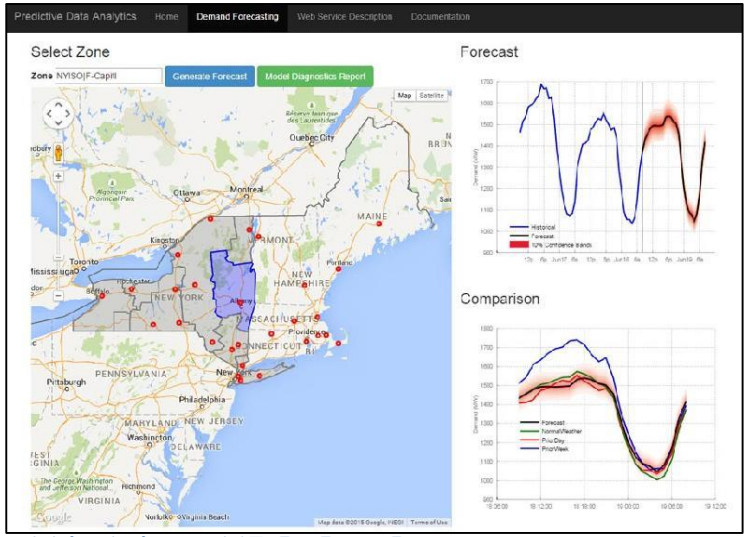
# 机器学习推动科技创新



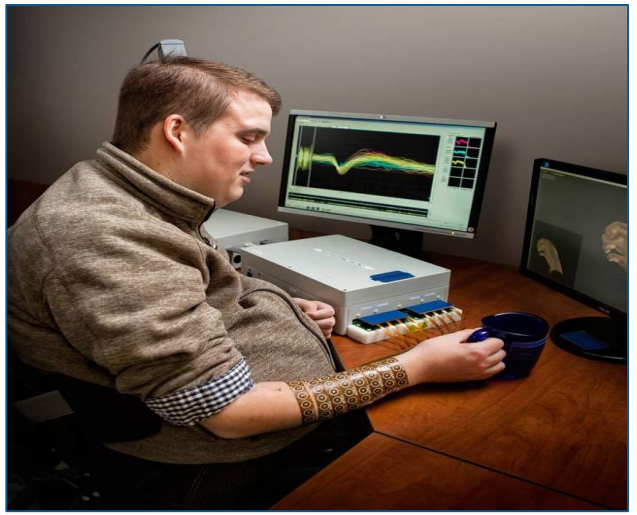
机器人模仿复杂人类行为




情绪分析在金融中的应用




电网负荷预测



恢复对瘫痪手臂的控制

# Battelle 神经旁路技术恢复了瘫痪者手臂和手的运动

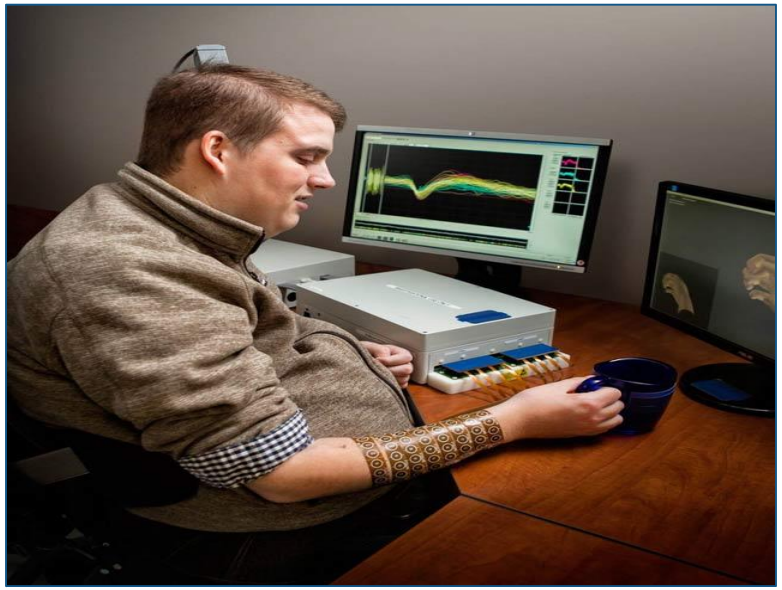
## 挑战
通过处理来自植入他脑组织的电极阵列的信号，恢复手臂和手的控制

## 使用产品
MATLAB + Wavelet Toolbox

## 方法和途径
- 用MATLAB对信号样本进行分析
- 利用小波生成紧凑的特征向量——**小波使研究人员能够从信号中提取重要信息进行分类**
- 应用机器学习对映射到运动的模式进行分类，并为神经肌肉电刺激器生成驱动信号

## 结果
- 对瘫痪手和手臂的控制恢复
- 达到实时处理性能

Link to full user story



患者 Battelle NeuroLife 系统

**“我们用 MATLAB 开发的算法让参与者对他的手臂和手有了基本的控制。在研究结束的时候，他可以抓住一个瓶子，倒出里面的东西，然后把它放下，拿起搅拌棒，接着开始搅拌。”**

**David Friedenberg**
**Battelle**

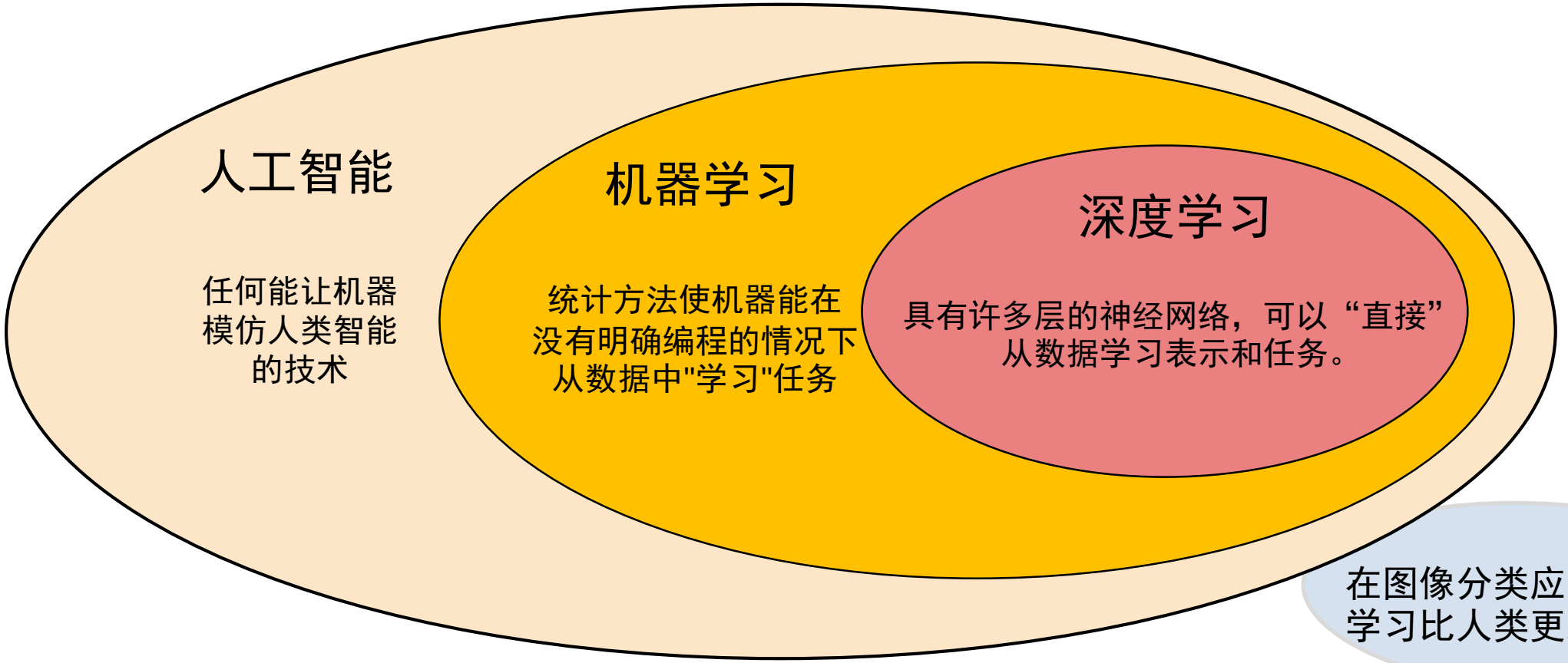
# 内容概要

机器学习工作流程及其面临的挑战
机器学习类型综述
心音分类器的开发
深度学习应用

## 关键要点

- 涵盖完整的工作流程（探索到部署）
- 是机器学习变得简单
- 支持深度学习

# 人工智能，机器学习和深度学习

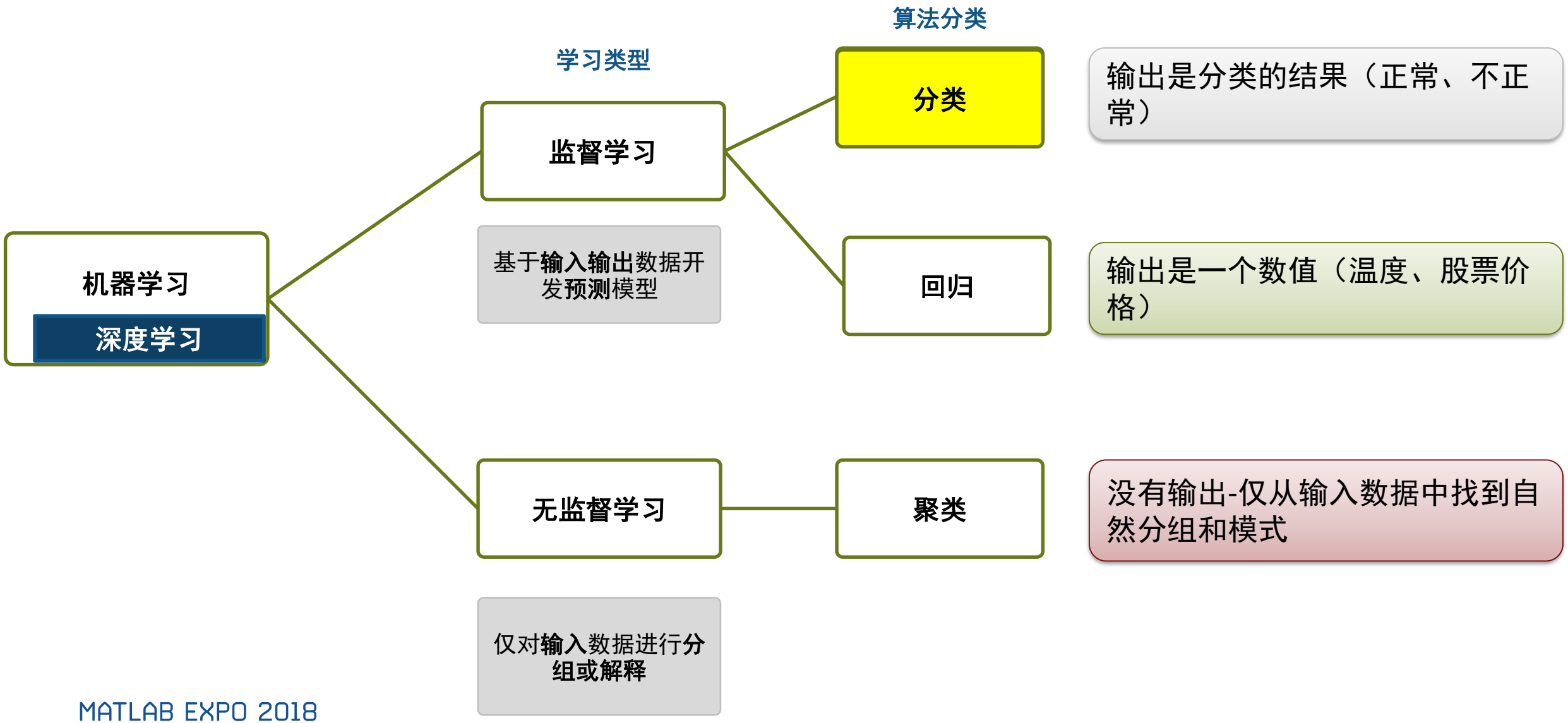



在图像分类应用中，深度学习比人类更准确

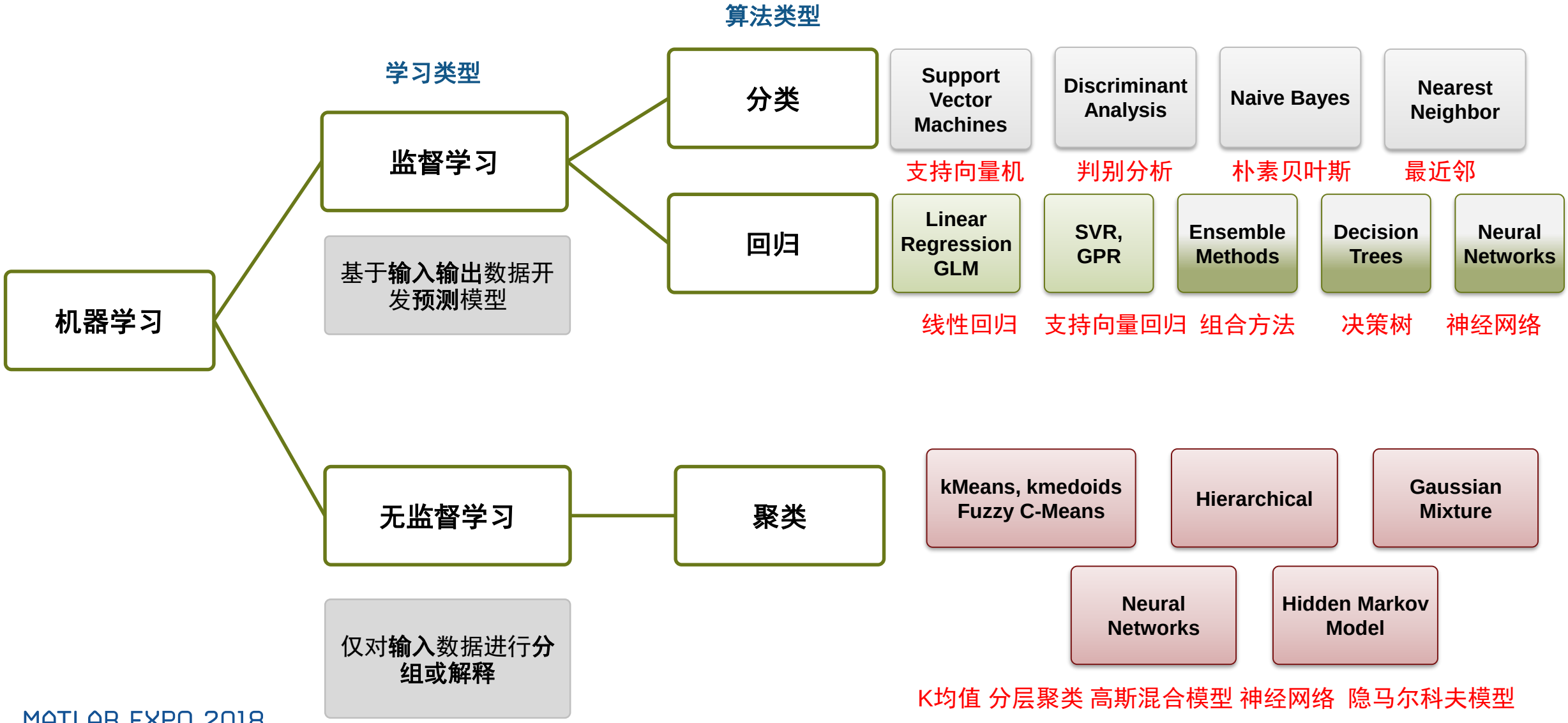
1950s　　1980s　　2015

FLOPS　　*Thousand*　　*Million*　　*Quadrillion*

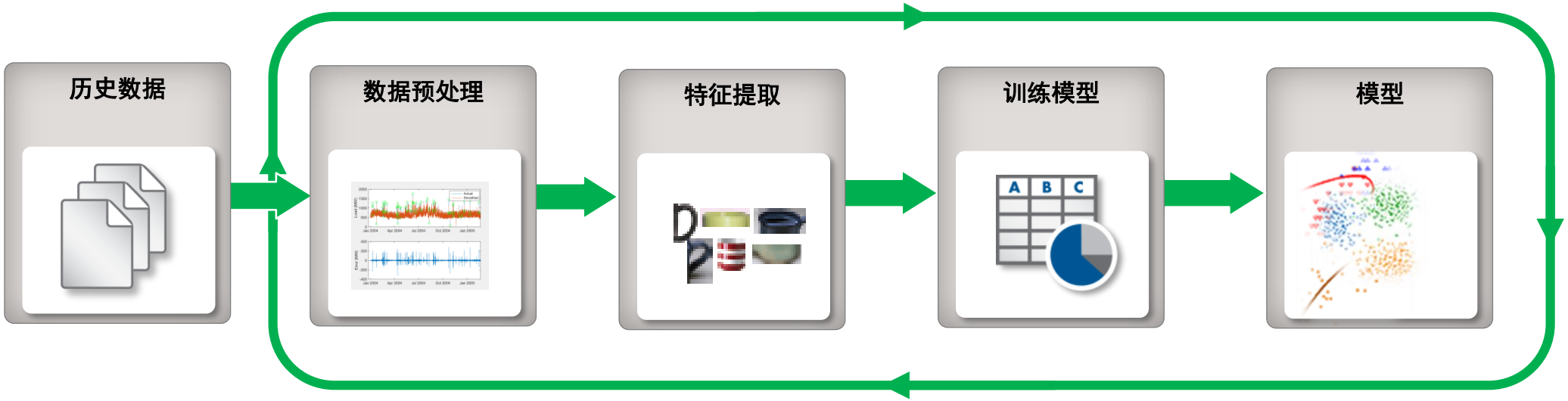
# 机器学习类型


算法分类
学习类型
机器学习
深度学习
监督学习
基于输入输出数据开发预测模型
分类
回归
输出是分类的结果（正常、不正常）
输出是一个数值（温度、股票价格）
无监督学习
仅对输入数据进行分组或解释
聚类
没有输出-仅从输入数据中找到自然分组和模式

# 机器学习分类


算法类型
学习类型
机器学习
监督学习
基于输入输出数据开发预测模型
分类
回归
Support Vector Machines
Discriminant Analysis
Naive Bayes
Nearest Neighbor
支持向量机
判别分析
朴素贝叶斯
最近邻
Linear Regression GLM
SVR, GPR
Ensemble Methods
Decision Trees
Neural Networks
线性回归
支持向量回归
组合方法
决策树
神经网络
无监督学习
仅对输入数据进行分组或解释
聚类
kMeans, kmedoids Fuzzy C-Means
Hierarchical
Gaussian Mixture
Neural Networks
Hidden Markov Model
K均值
分层聚类
高斯混合模型
神经网络
隐马尔科夫模型

# 机器学习流程

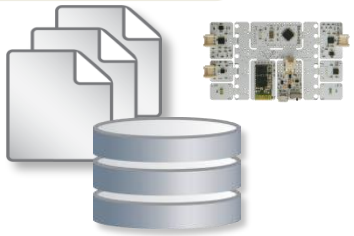




**训练:** 迭代直到找到最好的模型

# 开发机器学习应用面临的挑战

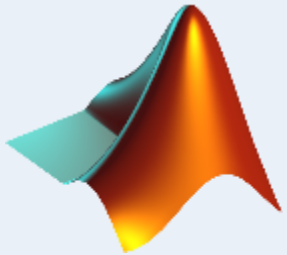
1. 数据访问

*各类传感器协议*



2. 探索及预处理

*各类数据*
*清理杂乱数据*
*模式探索*



3. 特征提取

*选择特征所需的领域知识*

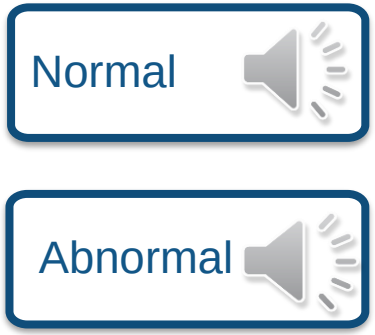
4. 创建模型

*多种算法*
*参数调整*

5. 部署

*不同的平台*
*规模/速度*

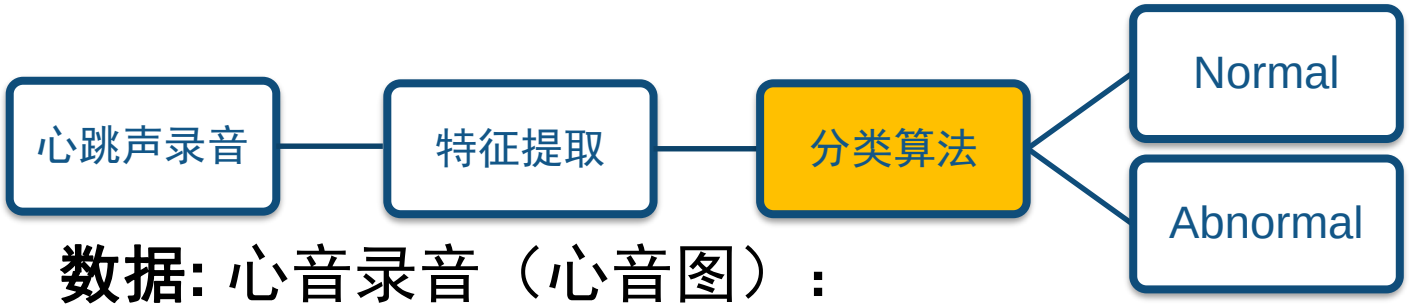
# 示例学习：心音分类器




**动机：**

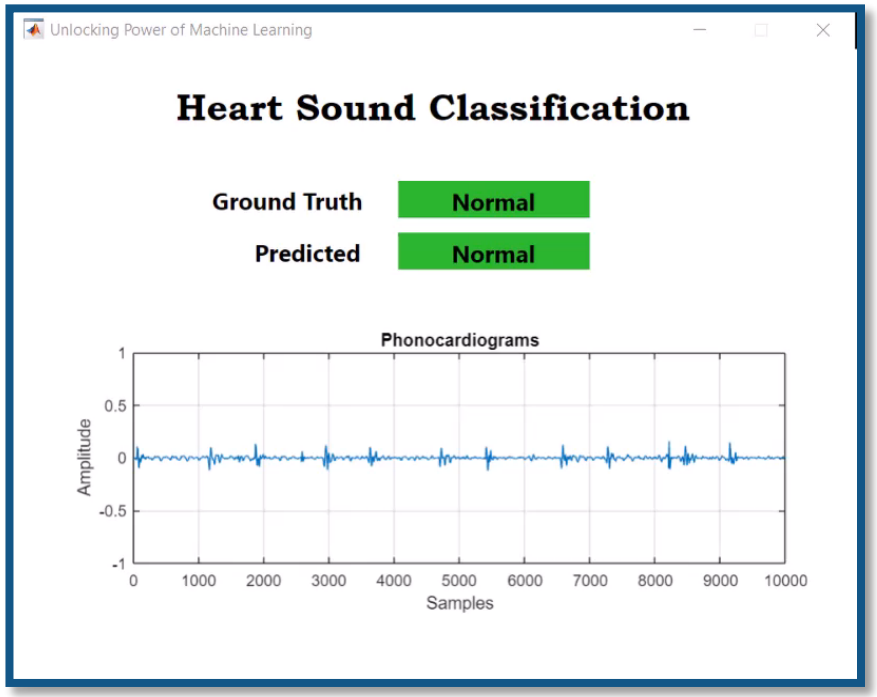
- 心跳声是复杂的信号
- 需要受过训练的临床医生诊断

**目标：开发分类器并部署到便携设备**




**数据: 心音录音（心音图）：**

- 来自 PhysioNet Challenge 2016
- 录音长度: 5 到 120 秒
- 训练数据: 3240 段录音 vs 测试数据：301 段录音
- 标签: Normal 或 Abnormal



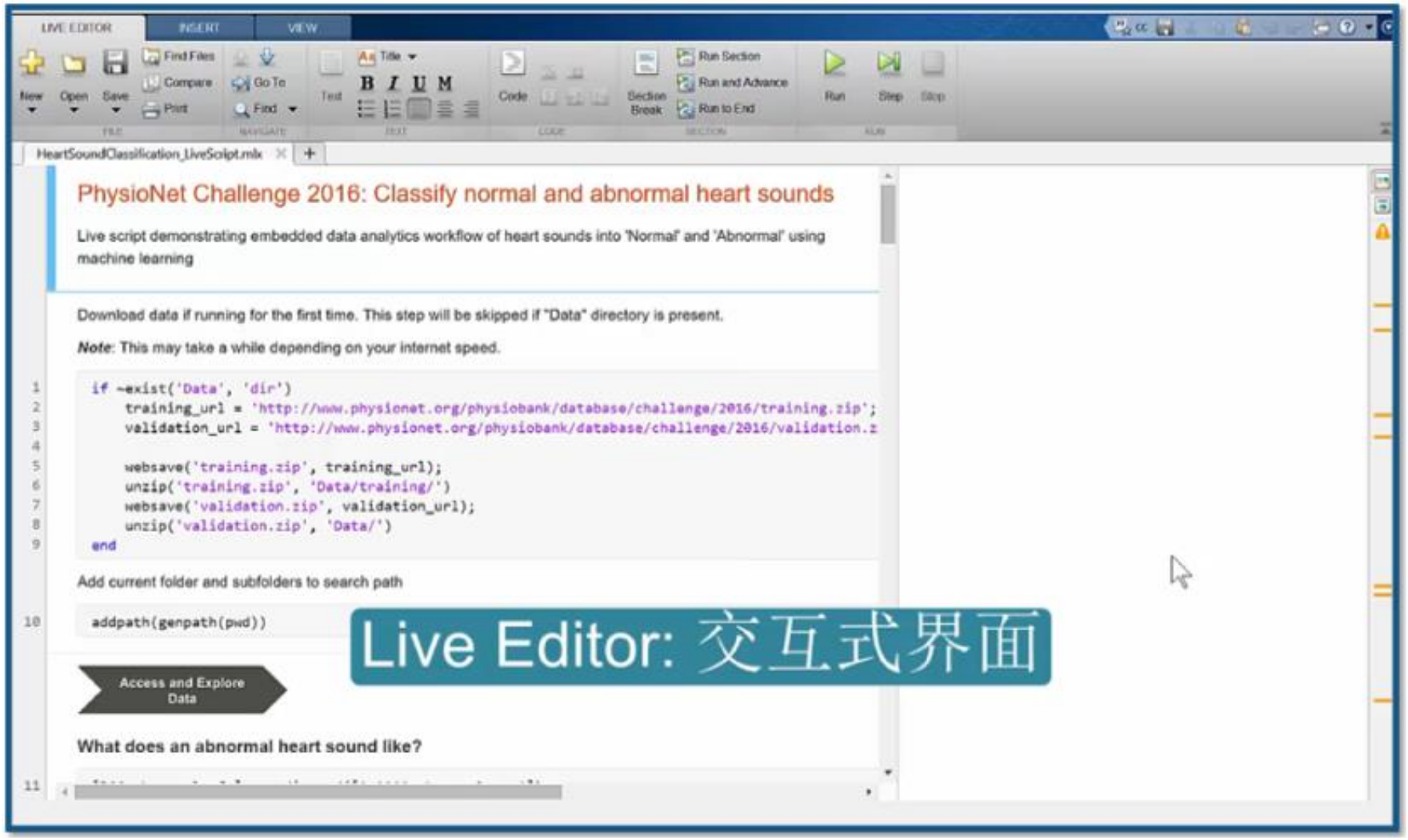
# 第一步：数据访问和探索

挑战：

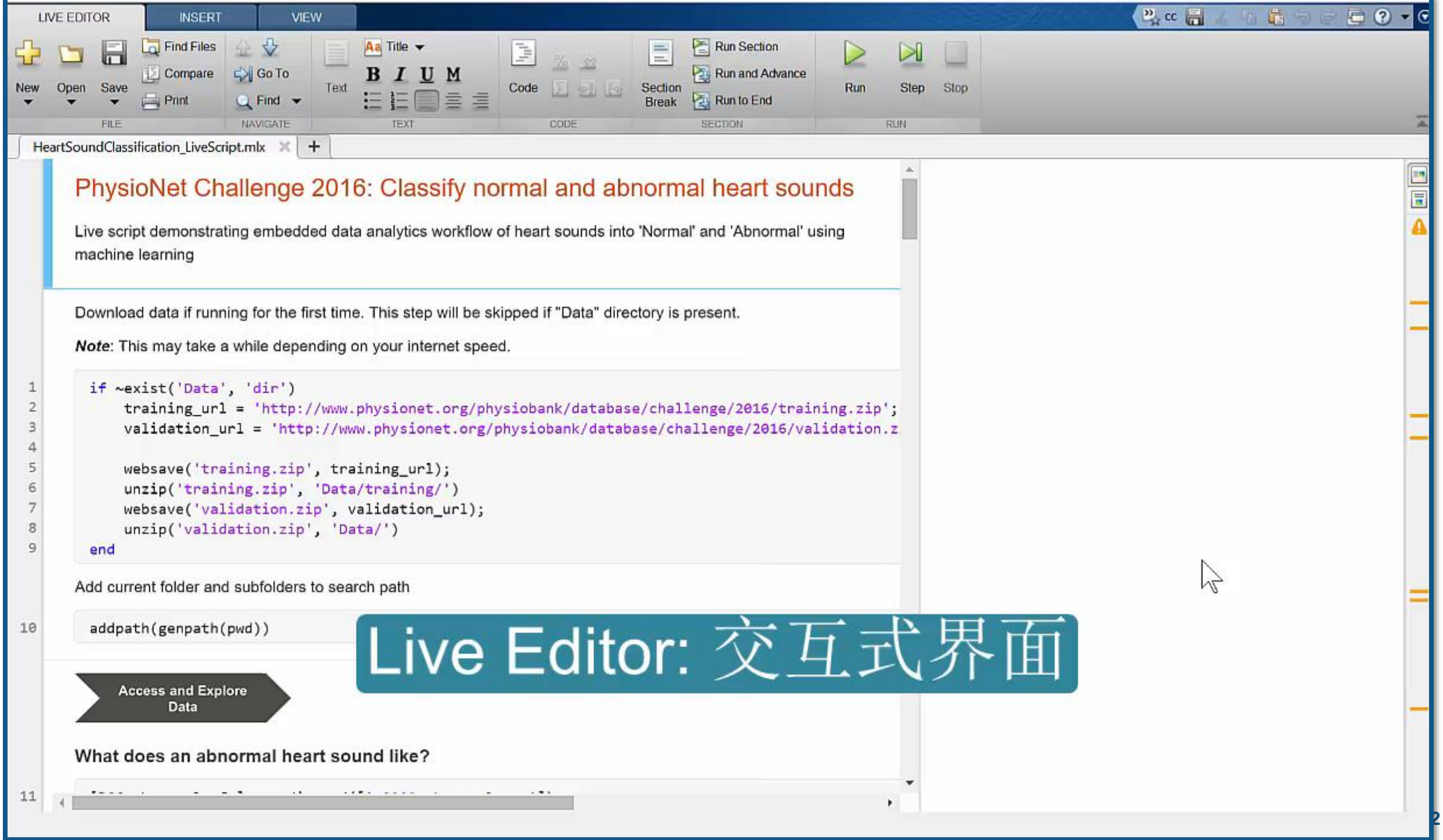
- 不同的采样率
- 信号管理
- 大规模数据集（“大数据”）

数据探索简单可行：

- 时域
- 频域
- 时频域




*信号分析仪：数据探索可视化*


LIVE EDITOR
INSERT
VIEW
New
Open
Save
Find Files
Compare
Print
Go To
Find
Text
Title
Code
Section Break
Run Section
Run and Advance
Run to End
Run
Step
Stop
FILE
NAVIGATE
TEXT
CODE
SECTION
RUN
HeartSoundClassification_LiveScript.mlx
PhysioNet Challenge 2016: Classify normal and abnormal heart sounds
Live script demonstrating embedded data analytics workflow of heart sounds into 'Normal' and 'Abnormal' using machine learning
Download data if running for the first time. This step will be skipped if "Data" directory is present.
Note: This may take a while depending on your internet speed.
if ~exist('Data', 'dir')
training_url = 'http://www.physionet.org/physiobank/database/challenge/2016/training.zip';
validation_url = 'http://www.physionet.org/physiobank/database/challenge/2016/validation.z
websave('training.zip', training_url);
unzip('training.zip', 'Data/training/')
websave('validation.zip', validation_url);
unzip('validation.zip', 'Data/')
end
Add current folder and subfolders to search path
addpath(genpath(pwd))
Access and Explore Data
What does an abnormal heart sound like?
Live Editor: 交互式界面

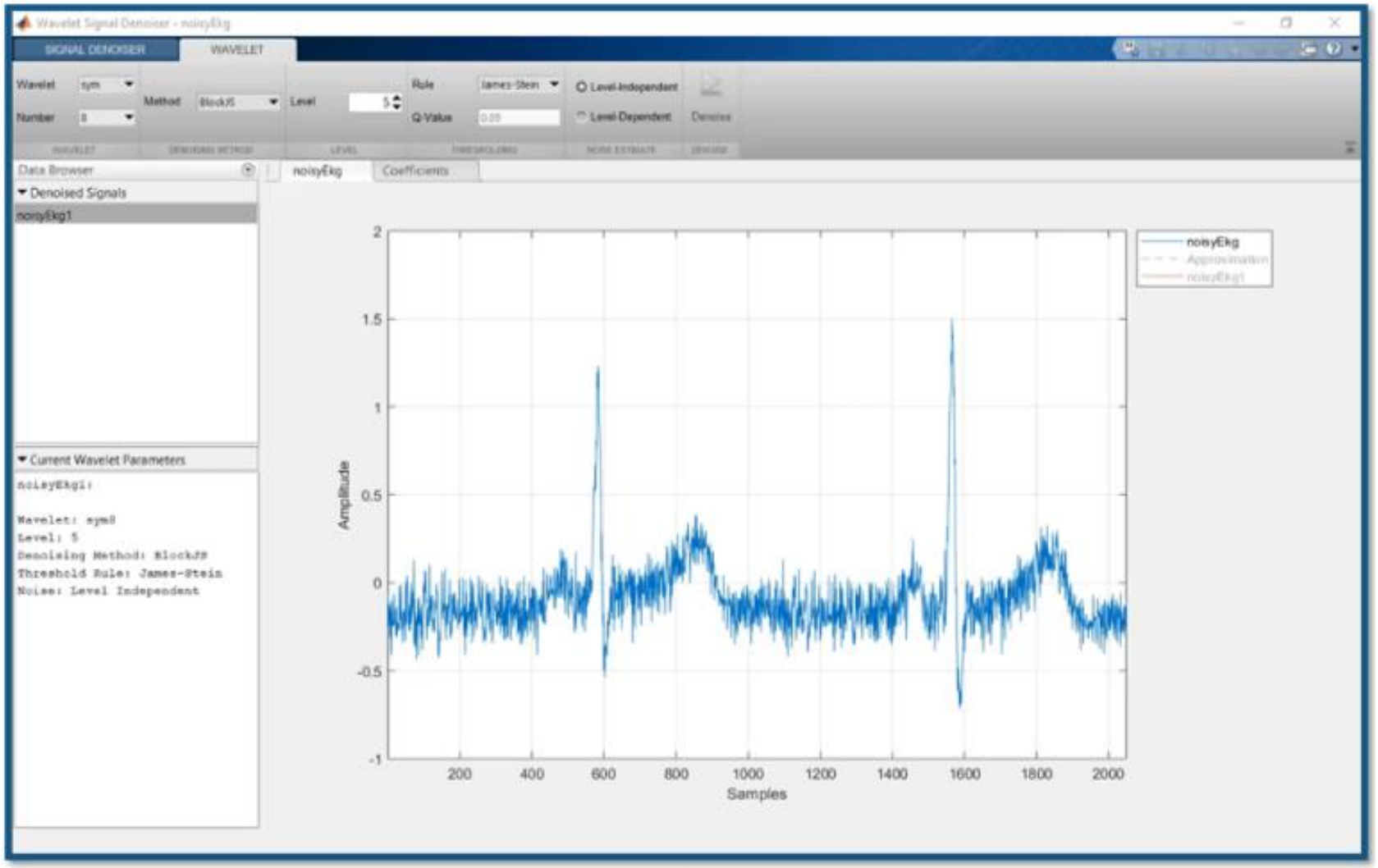
# 第二步：数据预处理

挑战：

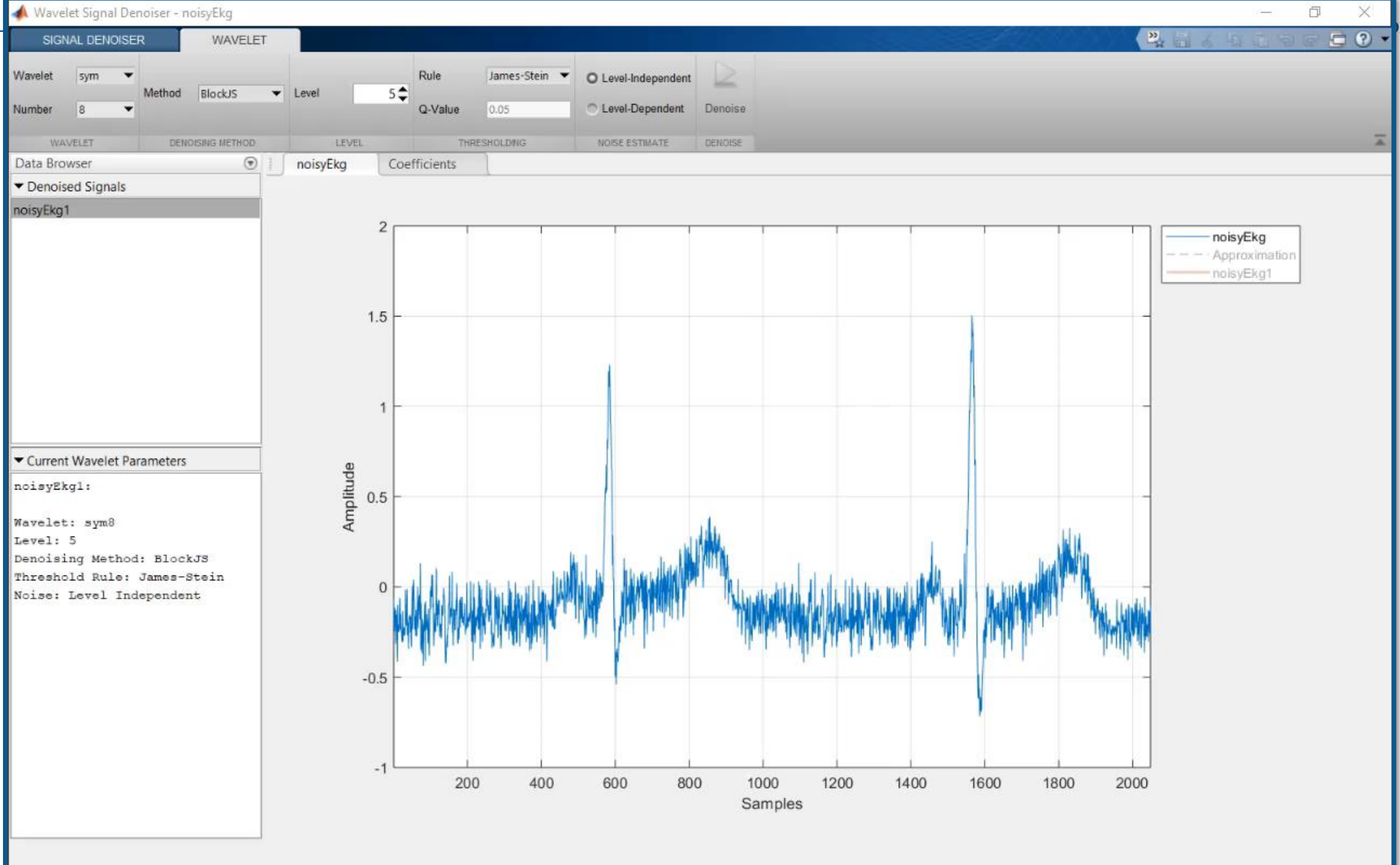
- 保留峰值特征
- 信号与噪声谱重叠

自动去噪声

- Wavelet Signal Denoiser
- MATLAB 代码生成




*无需编写任何代码进行信号预处理*


Wavelet Signal Denoiser - noisyEkg
SIGNAL DENOISER
WAVELET
Wavelet
sym
Number
8
Method
BlockJS
Level
5
Rule
James-Stein
Q-Value
0.05
Level-Independent
Level-Dependent
Denoise
WAVELET
DENOISING METHOD
LEVEL
THRESHOLDING
NOISE ESTIMATE
DENOISE
Data Browser
Denoised Signals
noisyEkg1
Current Wavelet Parameters
noisyEkg1:
Wavelet: sym8
Level: 5
Denoising Method: BlockJS
Threshold Rule: James-Stein
Noise: Level Independent
noisyEkg
Coefficients
noisyEkg
Approximation
noisyEkg1
Amplitude
Samples
2
1.5
1
0.5
0
-0.5
-1
200
400
600
800
1000
1200
1400
1600
1800
2000

# 数据预处理

*更简单的方法来清理凌乱的数据*

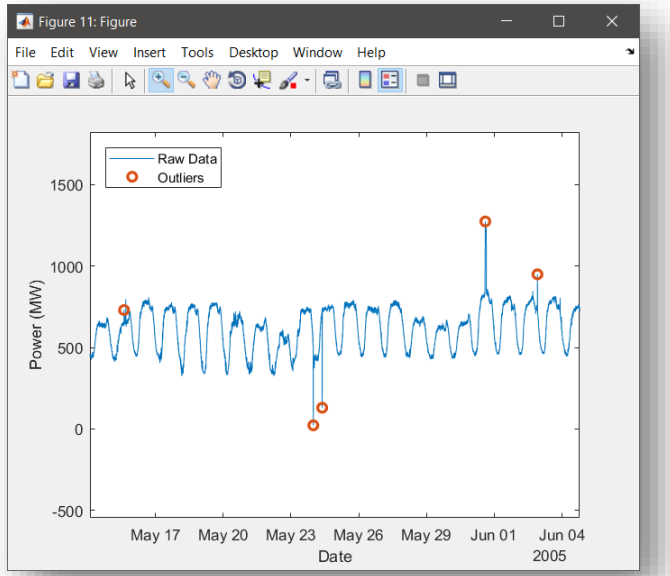
R2016b

- 使用 `*missing` 函数查找、填充和删除丢失的数据
- 可以使用累积统计函数忽略“NaNs”
- 使用 `replace`, `contains`, `endsWith` 等进行更简单的文本操作
- 使用 `smoothdata` 对数据进行滤波或局部回归以平滑噪声

R2017a

- 使用 `isoutlier` 和 `filloutliers` 处理离群值

R2017a



# 第三步：特征提取

挑战：

- 查找非平稳信号的特征
- 特征出现在不同的尺度上
- 特征选择

时域统计特征：

- 平均值、中值
- 标准差

音频信号频域特征：

- 主频率
- Mel频率倒谱系数（MFCC）
- 用小波进行倍频程分解

```
% Distribute computation across available processors by
parfor ipart = 1:n_parts
    % Get partition ipart of the datastore.
    subds = partition(training_fds, n_parts, ipart);

    % Extract features for the sub datastore
    feature_table = [feature_table; extractFeatures(subd

    % Display progress
    disp(['Part ' num2str(ipart) ' done.'])
end
save('FeatureTable', 'feature_table');
se
load('FeatureTable.mat');
```

特征提取

# 第四步：训练模型

挑战：

- 机器学习算法的知识
- 扩展到大规模数据集

App 中快速训练模型

- 定义交叉验证
- 尝试各种常用算法
- 性能分析：
  测试数据准确率 93%

无需重新编码即可扩展到大数据集：Tall Array

| 算法 | 预测速度 | 训练速度 | 内存使用 | 需要调优 | 一般评估 |
|---|---|---|---|---|---|
| 逻辑回归 (和线性 SVM) | 快 | 快 | 小 | 最小 | 擅长解决有线性决策边界的小问题 |
| 决策树 | 快 | 快 | 小 | 有些 | 通用性很好，但容易过度拟合 |
| (非线性) SVM (和逻辑回归) | 慢 | 慢 | 中等 | 有些 | 擅长解决许多二进制问题，能很好地处理高维度数据 |
| 最近邻 | 适中 | 最小 | 中等 | 最小 | 精度较低，但易于使用和解释 |
| 朴素贝叶斯 | 快 | 快 | 中等 | 有些 | 广泛用于文本，包括垃圾邮件过滤 |
| 集成学习 | 适中 | 慢 | 差异大 | 有些 | 对于中小规模的数据集，精度高，性能好 |
| 神经网络 | 适中 | 慢 | 中到大 | 很多 | 普遍用于分类、压缩、识别和预测 |

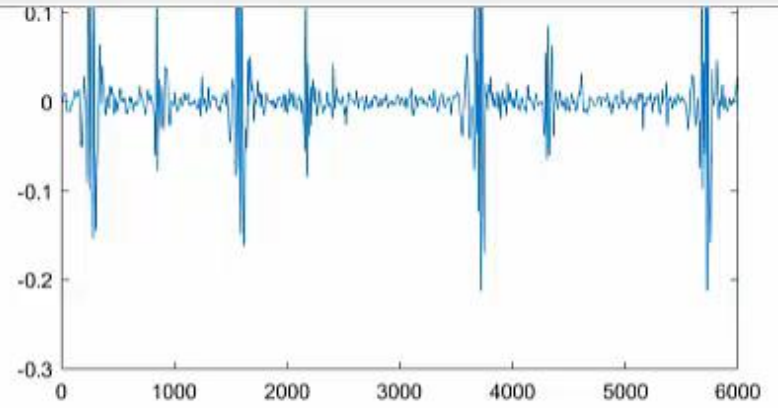
LIVE EDITOR | INSERT | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Text | Normal | Code | Section Break | Run Section | Run and Advance | Run to End | Run | Step | Stop

FILE | NAVIGATE | TEXT | CODE | SECTION | RUN

HeartSoundClassification_LiveScript.mlx | extractFeatures.m

```
56
57          % Extract features for the sub datastore
58          feature_table = [feature_table; extractFeatures(subds, win_len, win_overlap,
59
60          % Display progress
61          disp(['Part ' num2str(ipart) ' done.'])
62      end
63      save('FeatureTable', 'feature_table');
64  else
65      load('FeatureTable.mat');
66  end
67
68  % Take a look at the feature table
69  disp(feature_table(1:5,:))
```




| WAVFEAT17 | WAVFEAT18 | WAVFEAT19 | WAVFEAT20 | WAVFEAT21 |
|---|---|---|---|---|
| 0.24148 | 0.26671 | 0.012951 | 4.1958e-05 | 3.5735e-08 |
| 0.30883 | 0.086174 | 0.0087586 | 4.8264e-05 | 9.8086e-09 |
| 0.40704 | 0.13199 | 0.014097 | 0.00035801 | 7.3899e-07 |
| 0.47754 | 0.17483 | 0.005399 | 2.1398e-05 | 8.771e-09 |
| 0.33189 | 0.12301 | 0.023209 | 5.4791e-05 | 6.5493e-09 |

Develop Predictive Models

### Train, compare and select classifier

Use the Classification Learner App to interactively train, compare and select classifiers.

```
70  classificatio
```

打开 Classification Learninger App

Develop Predictive Models

### Split data into training and testing sets.

# 第四步：模型优化

挑战：

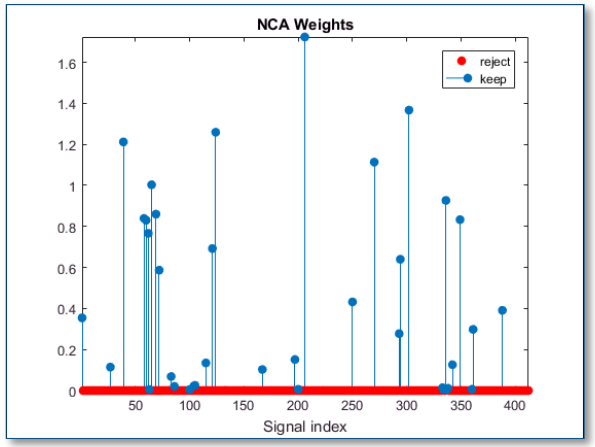
- 手动参数调整繁琐
- 额外改进

模型迭代优化

- 参数贝叶斯优化
- 性能分析可视化
- 调整不平衡（数据或者错误分类的严重程度）
- 使用自动特征选择，减少模型规模

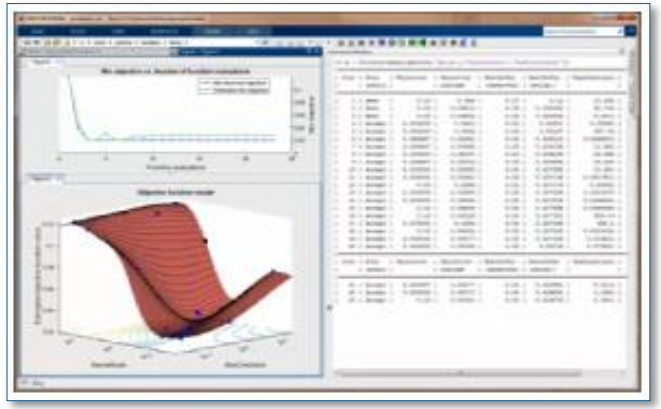
| Class | Distribution |
|---|---|
| Normal | 75% |
| Abnormal | 25% |

自动选择最佳“特征”

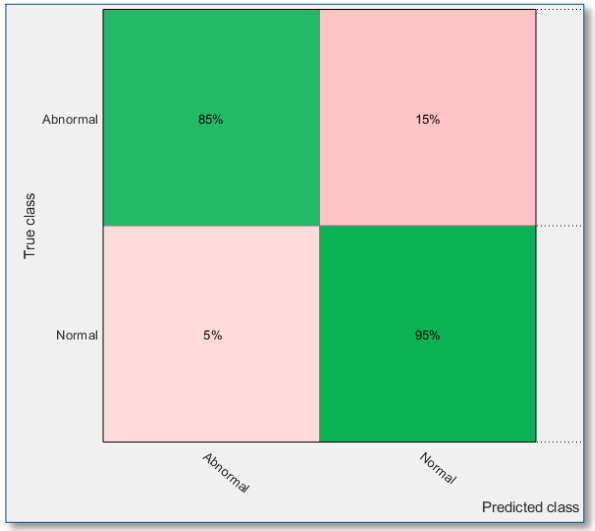



NCA：近邻成分分析

自动微调学习参数



超参数调优




混淆矩阵

LIVE EDITOR | INSERT | VIEW

New | Open | Save | Find Files | Compare | Print — FILE
Go To | Find — NAVIGATE
Text | Normal | B I U M — TEXT
Code — CODE
Section Break | Run Section | Run and Advance | Run to End — SECTION
Run | Step | Stop — RUN

HeartSoundClassification_LiveScript.mlx * | extractFeatures.m

```
81  grpstats_training = grpstats(training_set, 'class', 'mean');
82  disp(grpstats_training(:,'GroupCount'))
```

```
                GroupCount
                __________

Abnormal        1579
Normal          4929
```

**Develop Predictive Models**

### Train the classifier with misclassification cost

Train the classifier programmatically. To compensate for fewer 'Abnormal' observations, assign higher misclassification cost with the 'Abnormal' class. Perform hyperparameter tuning by using Bayesian Optimization to find optimal values for model parameters.

```
83  % Assign higher cost for misclassification of abnormal heart sounds
84  C = [0, 10; 1, 0];
85
86  % Create a random sub sample (to speed up training) from the training set
87  % subsample = randi([1 height(training_set)], round(height(training_set)/4), 1);
88  subsample = [1:height(training_set)];
89
90  % Create a 5-fold cross-validation set from training data
91  cvp = cvpartition(length(subsample),'KFold',5);
92
93  if ~exist('TrainedSVMModel.mat')
94      opts = struct('Optimizer','bayesopt','Sh
95          'AcquisitionFunctionName','expected-
96
97      rng(1)
98      trained_model = fitcsvm(training_set(subsample,:),'class','KernelFunction','rbf',
99          'OptimizeHyperparameters','auto','HyperparameterOptimizationOptions',opts)
100     save('TrainedSVMModel', 'trained_model');
101
```

引入“代价函数”

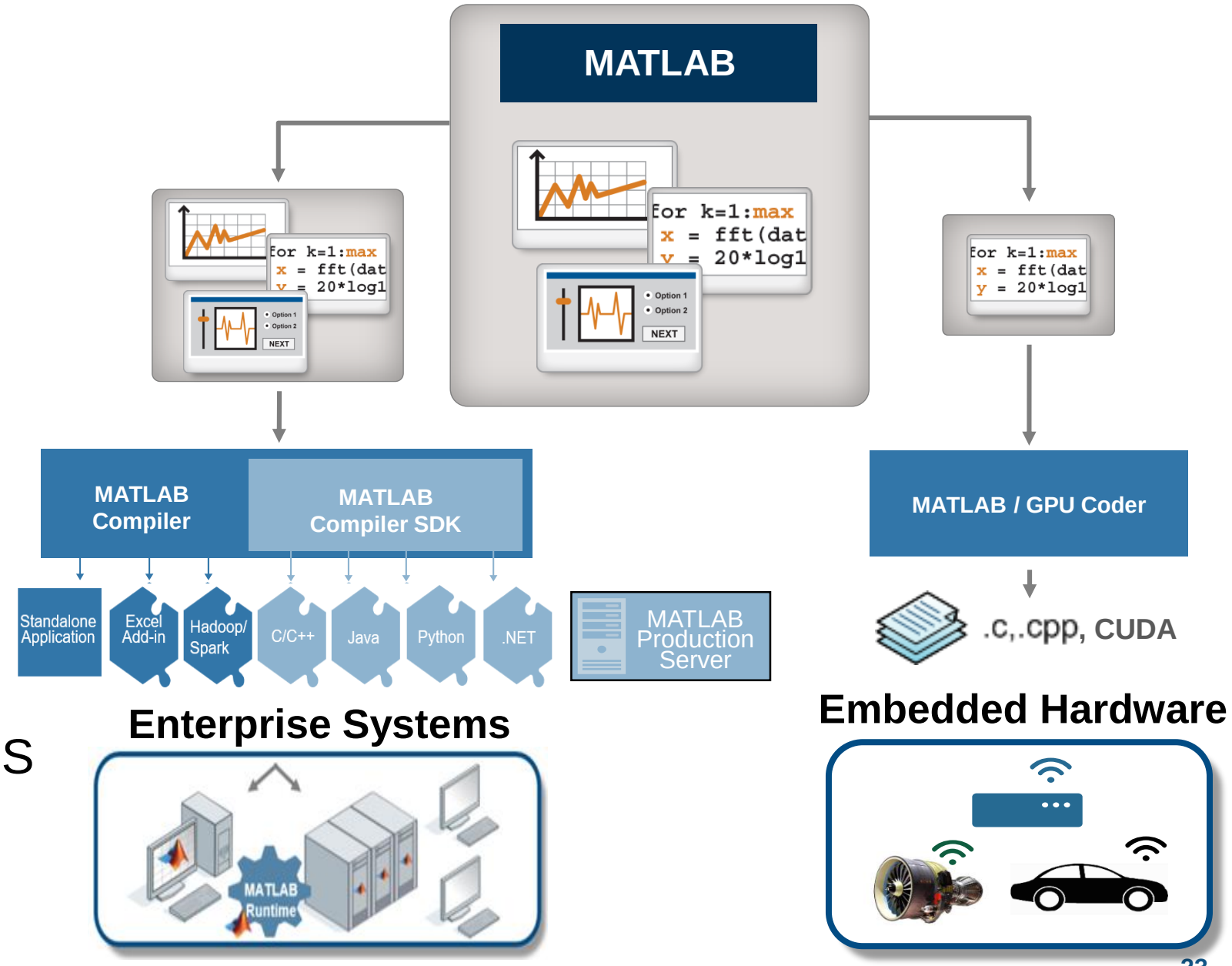
# 第五步：部署

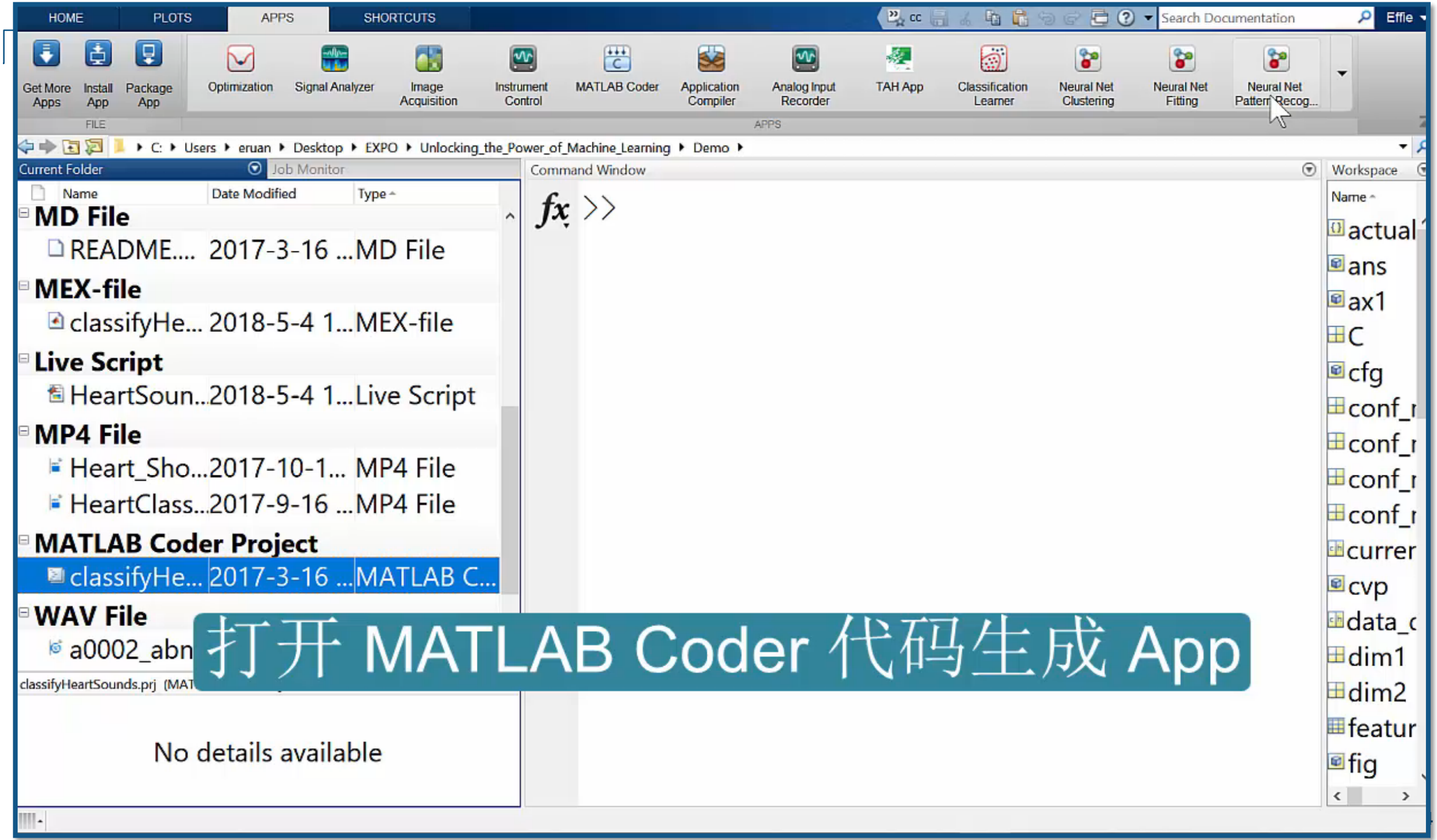
挑战：

- 不同的目标平台
- 硬件需求
  (规模，速度，定点等等)

部署选项：

- 为嵌入式系统生成代码
  （C，HDL，PLC）
- 编译 MATLAB，使用 MPS
  进行系统扩展





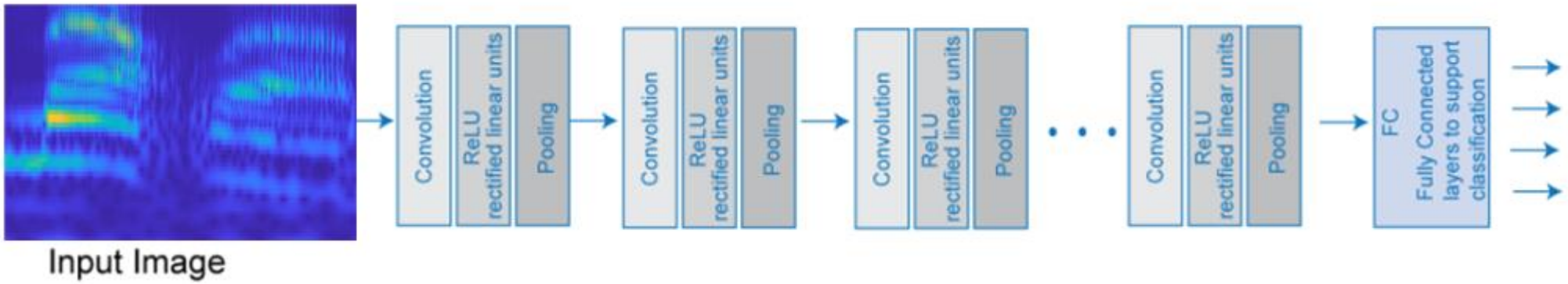

# 打开 MATLAB Coder 代码生成 App

# 基于信号的深度学习

## 使用多层神经网络进行监督分类

### 1. 卷积神经网络 (CNN)

- 多功能且灵活的深度学习方法
- 应用于经过时频转换的信号




### 2. 长短期记忆网络 (LSTM)

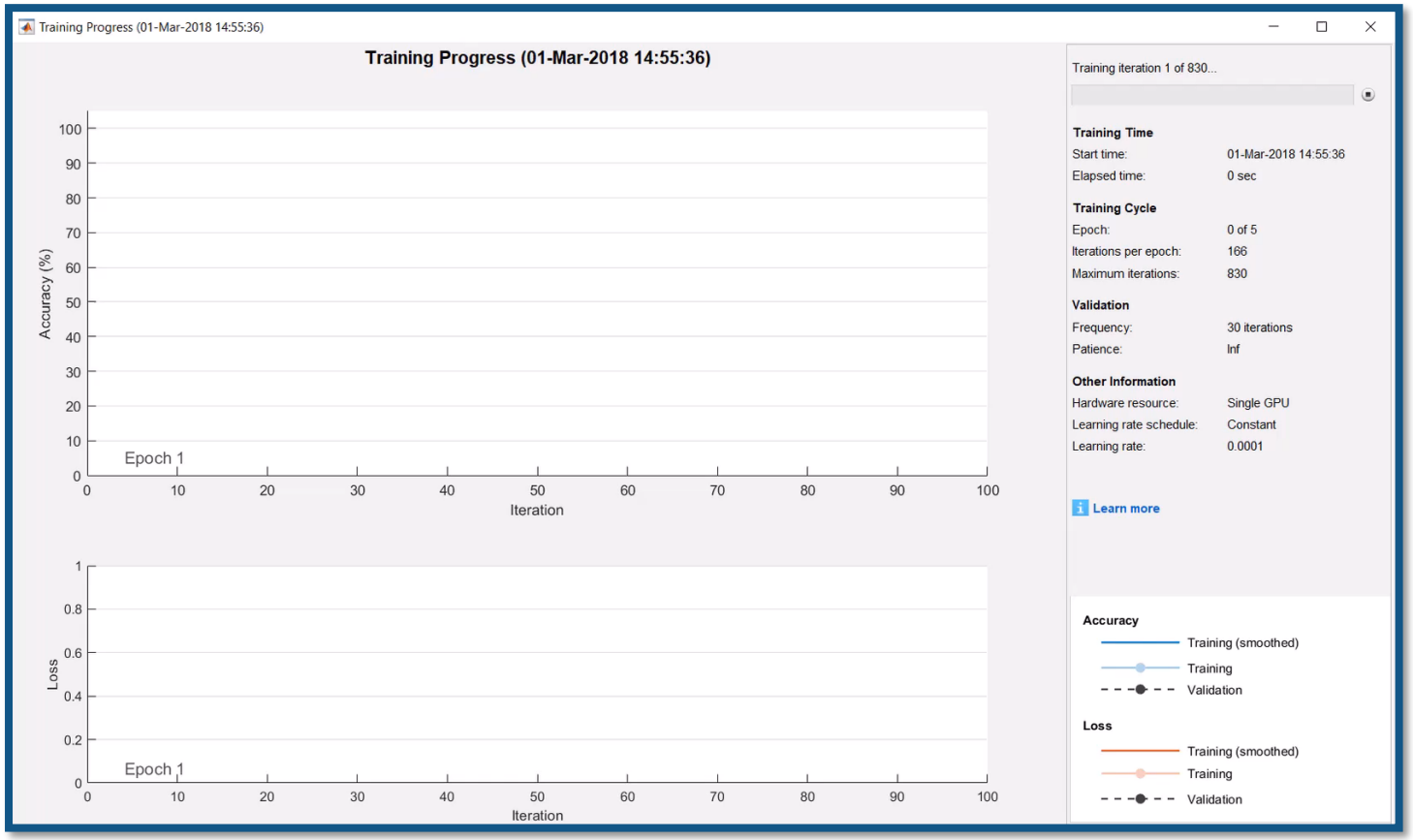
# 使用神经网络实现心音信号分类

## 步骤

- 信号 ➔ 时频表示
- 连续小波转换
- 利用 GoogleNet 进行迁移学习

## 结果

- 准确度达到 90%
- 只需 10 行代码



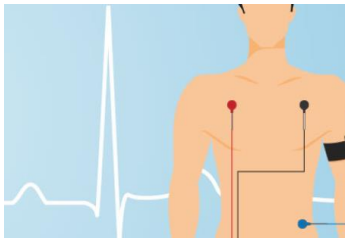

*深度学习训练*

R2017b

# 简要回顾：使机器学习变得容易

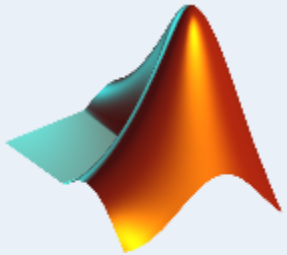
1. 数据访问

*支持工业传感器、手机等*



2. 探索及预处理

*可视化探索*



3. 特征提取

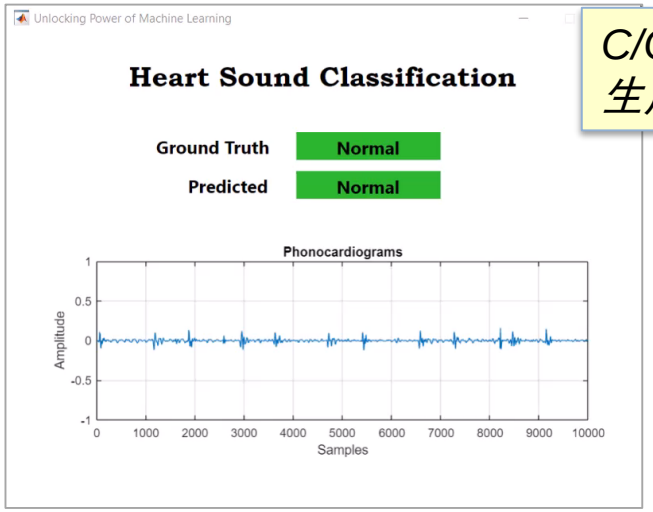
*小波*
*特征选择*

4. 创建模型

*快速比较 App 中的模型*
*自动参数调整*
*探索深度学习*

5. 部署

*C/CUDA 代码自动生成*

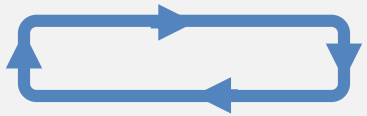


# 开发机器学习应用面临的挑战

## 起步艰难

| 步骤 | 挑战 |
| --- | --- |
| 访问、探索和分析数据 | **数据多样性**<br>数字、图像、信号、文本 —— 并不都是表格类型 |

# 开发机器学习应用面临的挑战

| 步骤 | 挑战 |
| --- | --- |
| 访问、探索和分析数据 | **数据多样性**<br>数字、图像、信号、文本 —— 并不都是表格类型 |
| 数据预处理 | **缺少特定领域的工具**<br>滤波与特征提取<br>特征选择与转换 |
| 训练模型 | **耗时**<br>训练多个模型已选取“最优” |
| 模型性能评价 | **避免陷阱**<br>过拟合<br>速度 – 准确度 – 复杂度权衡 |
| 迭代 |  |

# 关键要点

## 赋予工程师数据科学的力量！

- 涵盖完整的工作流程（探索到部署）
- 机器学习变得简单
- 支持深度学习

# 学习更多

查看 Battelle's "NeuroLife" 完整客户案例

从 File Exchange 下载 Heart Sounds Classification 应用

观看录制视频 "Machine Learning Using Heart Sound Classification"

阅读：

- Machine Learning with MATLAB
- What is Deep Learning?