MathWorks

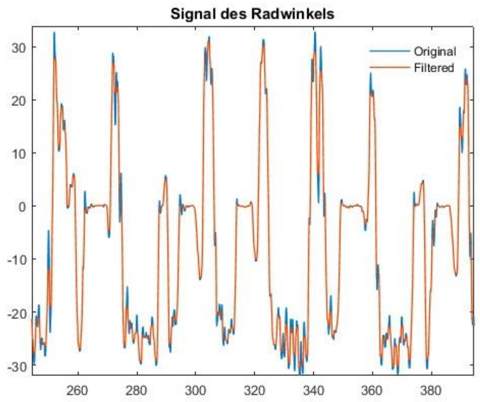
AUTOMOTIVE ENGINEERING CONFERENCE 2020

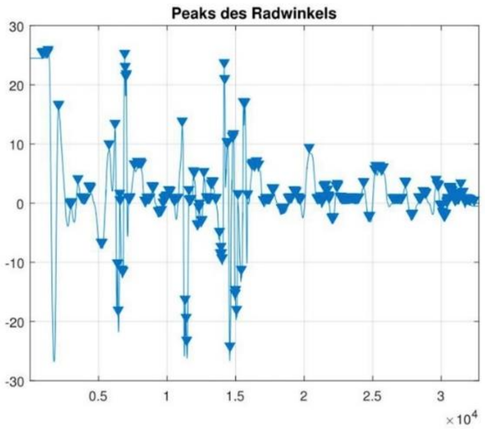
# Embedded Machine Learning:

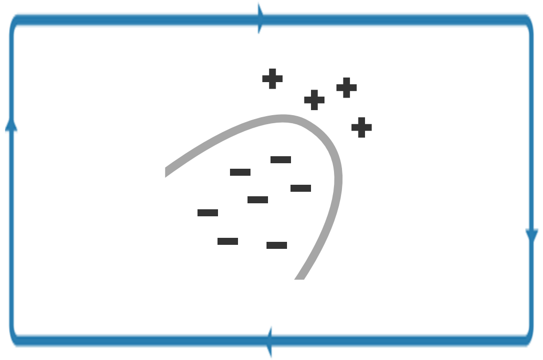
## Enabling Workflows for Edge Devices

Gokhan Atinc

# BMW designs, tests and deploys data-driven systems that enhance vehicles’ capabilities using MATLAB and Simulink



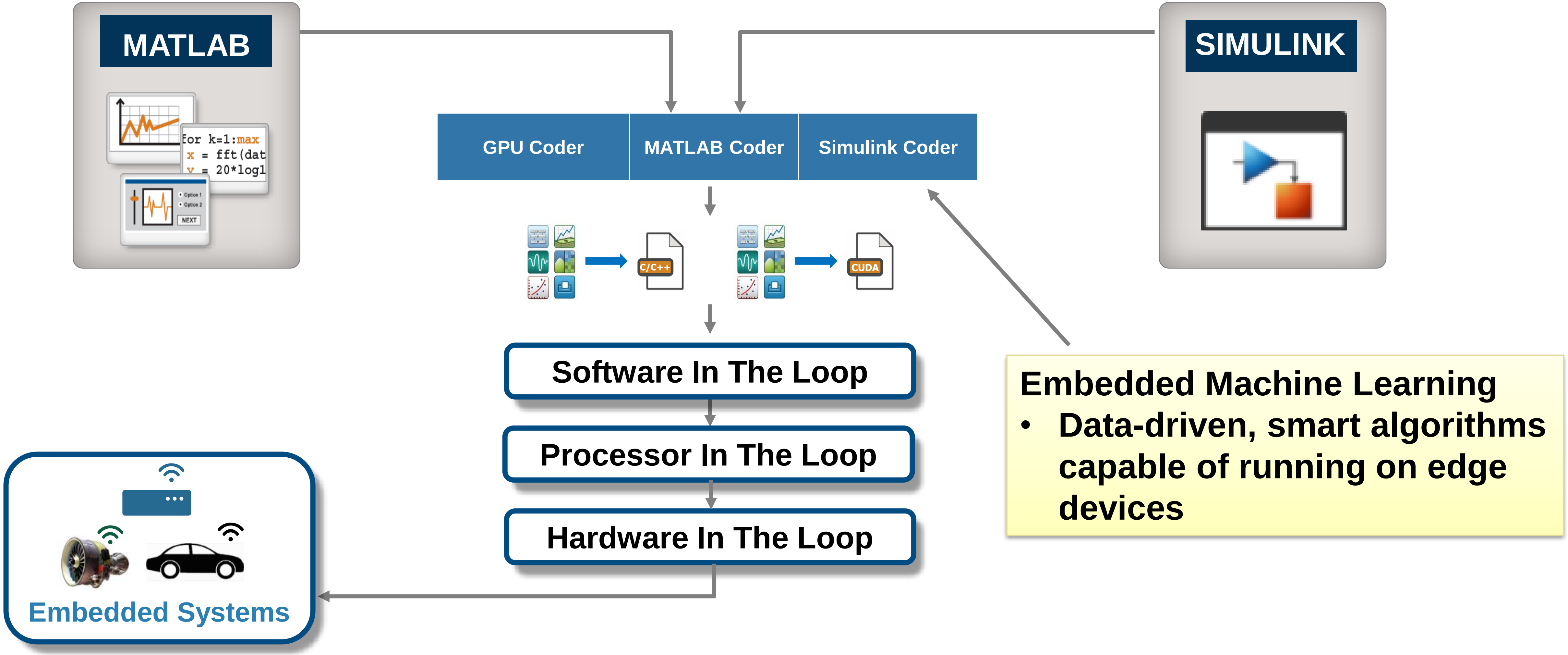




> 95% accuracy

Full Story: https://www.mathworks.com/company/newsletters/articles/detecting-oversteering-in-bmw-automobiles-with-machine-learning.html

# MathWorks provides embedded machine learning workflows that integrate nicely with Model-Based Design



MATLAB

SIMULINK

GPU Coder

MATLAB Coder

Simulink Coder

Software In The Loop

Processor In The Loop

Hardware In The Loop

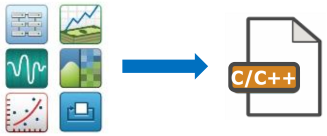
Embedded Systems

**Embedded Machine Learning**

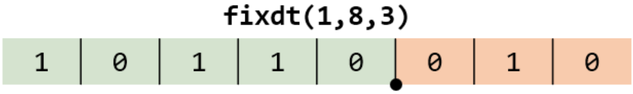
- Data-driven, smart algorithms capable of running on edge devices

# Machine learning algorithms are supported for a variety of embedded systems workflows

***Deploy machine learning models in MATLAB & Simulink***



***Deploy fixed-point machine learning models***

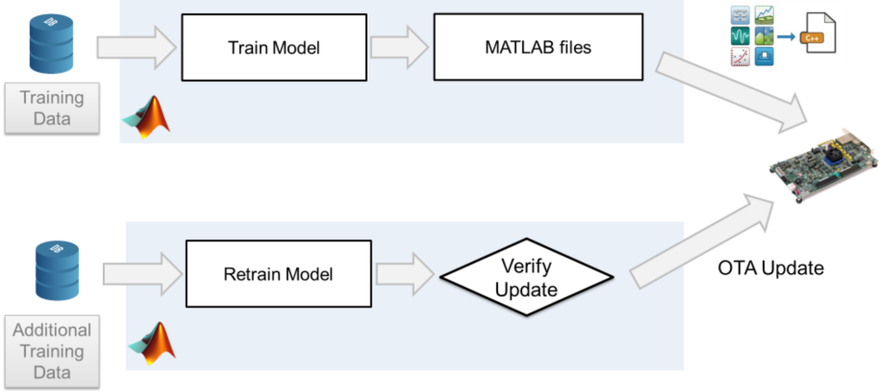


fixdt(1,8,3)

| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Real world value: -9.75

***In-place modification of deployed models***



Training Data

Train Model

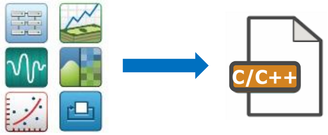
MATLAB files

Additional Training Data

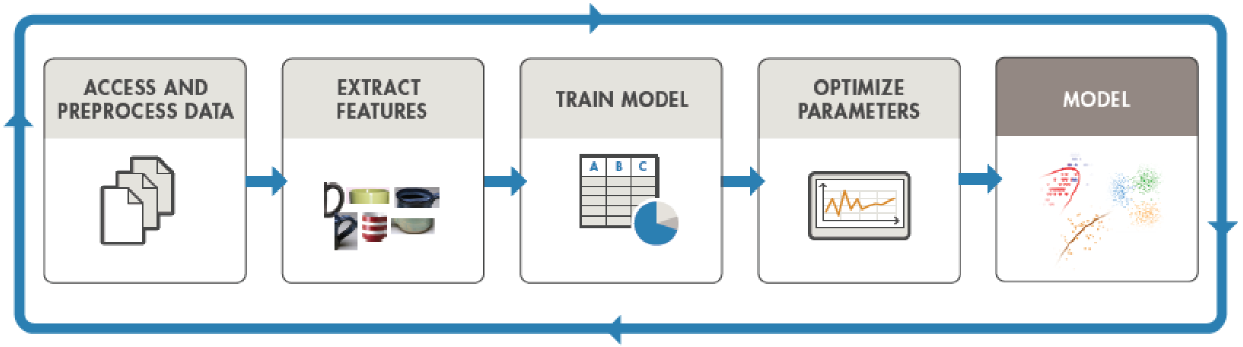
Retrain Model

Verify Update

OTA Update



# Learner apps provide convenient ways to compare and iterate over different machine learning algorithms



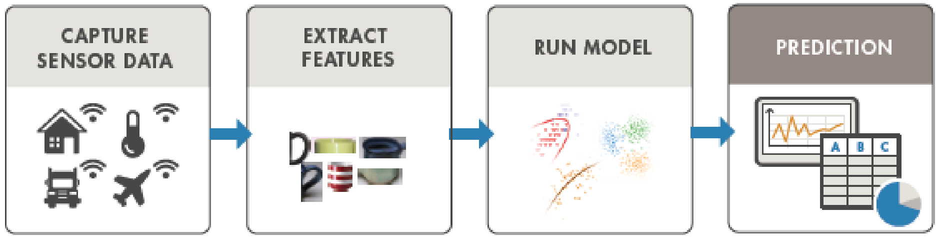
ACCESS AND PREPROCESS DATA

EXTRACT FEATURES

TRAIN MODEL

OPTIMIZE PARAMETERS

MODEL



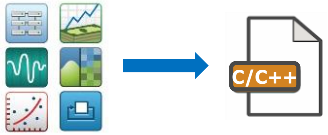
CAPTURE SENSOR DATA

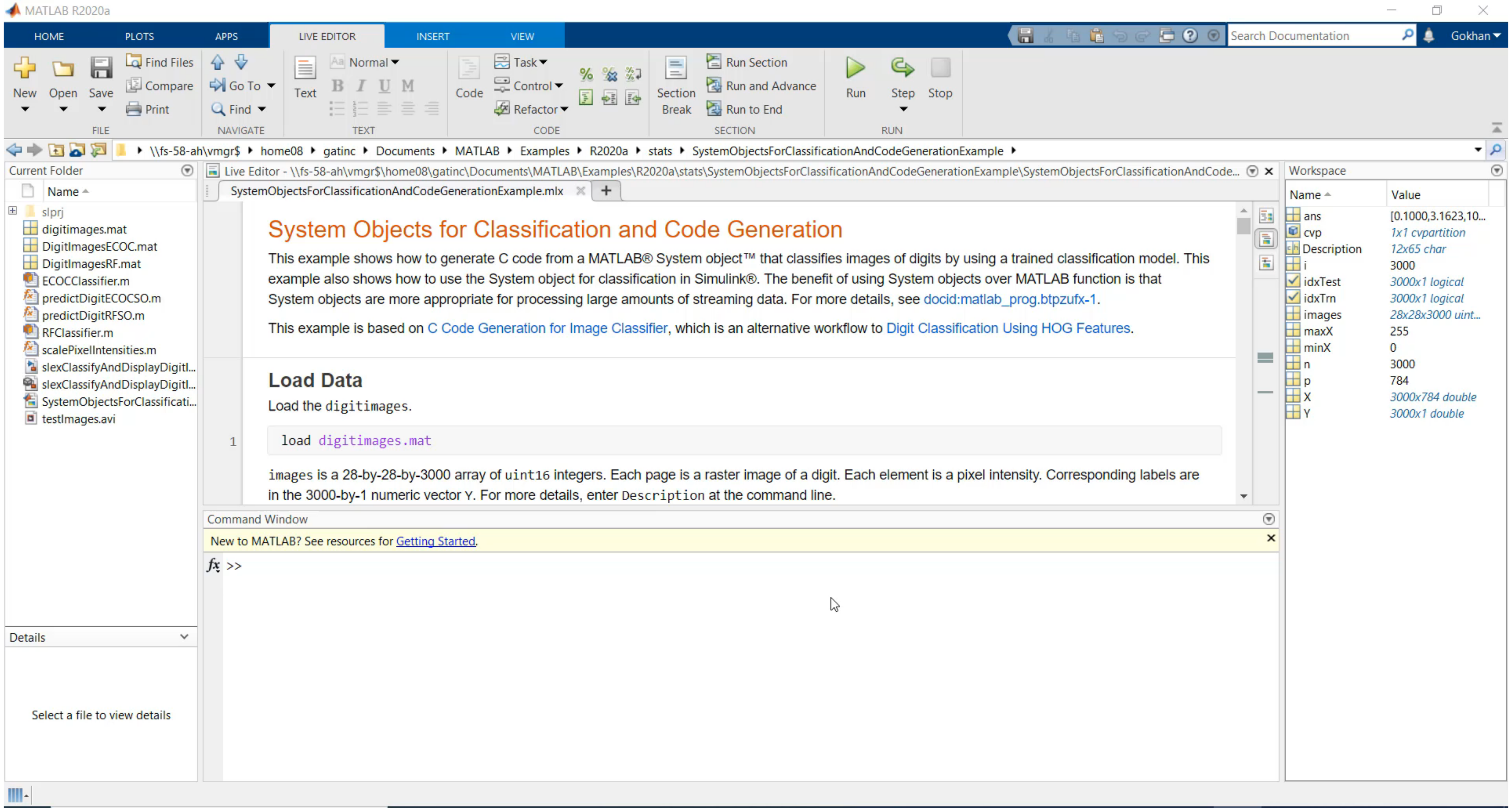
EXTRACT FEATURES

RUN MODEL

PREDICTION

# Classification Learner App demonstration





MATLAB R2020a
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
Search Documentation
Gokhan
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Text
Normal
TEXT
Code
Task
Control
Refactor
CODE
Section Break
Run Section
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
\\fs-58-ah\vmgr$ ▸ home08 ▸ gatinc ▸ Documents ▸ MATLAB ▸ Examples ▸ R2020a ▸ stats ▸ SystemObjectsForClassificationAndCodeGenerationExample ▸
Current Folder
Name
Live Editor - \\fs-58-ah\vmgr$\home08\gatinc\Documents\MATLAB\Examples\R2020a\stats\SystemObjectsForClassificationAndCodeGenerationExample\SystemObjectsForClassificationAndCode...
SystemObjectsForClassificationAndCodeGenerationExample.mlx
Workspace
Command Window
Details

Current Folder:

- slprj
- digitimages.mat
- DigitImagesECOC.mat
- DigitImagesRF.mat
- ECOCClassifier.m
- predictDigitECOCSO.m
- predictDigitRFSO.m
- RFClassifier.m
- scalePixelIntensities.m
- slexClassifyAndDisplayDigitI...
- slexClassifyAndDisplayDigitI...
- SystemObjectsForClassificati...
- testImages.avi

Select a file to view details

## System Objects for Classification and Code Generation

This example shows how to generate C code from a MATLAB® System object™ that classifies images of digits by using a trained classification model. This example also shows how to use the System object for classification in Simulink®. The benefit of using System objects over MATLAB function is that System objects are more appropriate for processing large amounts of streaming data. For more details, see docid:matlab_prog.btpzufx-1.

This example is based on C Code Generation for Image Classifier, which is an alternative workflow to Digit Classification Using HOG Features.

### Load Data

Load the `digitimages`.

```
load digitimages.mat
```

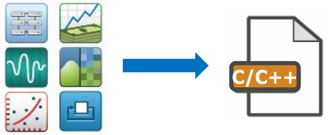
`images` is a 28-by-28-by-3000 array of `uint16` integers. Each page is a raster image of a digit. Each element is a pixel intensity. Corresponding labels are in the 3000-by-1 numeric vector `Y`. For more details, enter `Description` at the command line.

| Name | Value |
|---|---|
| ans | [0.1000,3.1623,10... |
| cvp | *1x1 cvpartition* |
| Description | *12x65 char* |
| i | 3000 |
| idxTest | *3000x1 logical* |
| idxTrn | *3000x1 logical* |
| images | *28x28x3000 uint...* |
| maxX | 255 |
| minX | 0 |
| n | 3000 |
| p | 784 |
| X | *3000x784 double* |
| Y | *3000x1 double* |

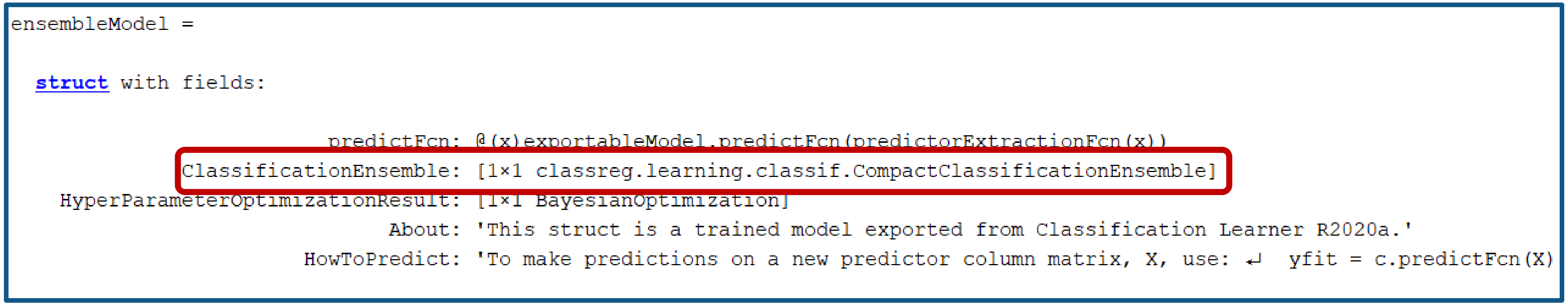
New to MATLAB? See resources for Getting Started.

```
fx >>
```

# Models trained with Learner App can be saved for deployment

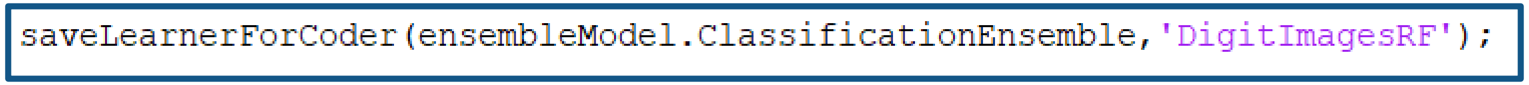


## Extract Trained Model



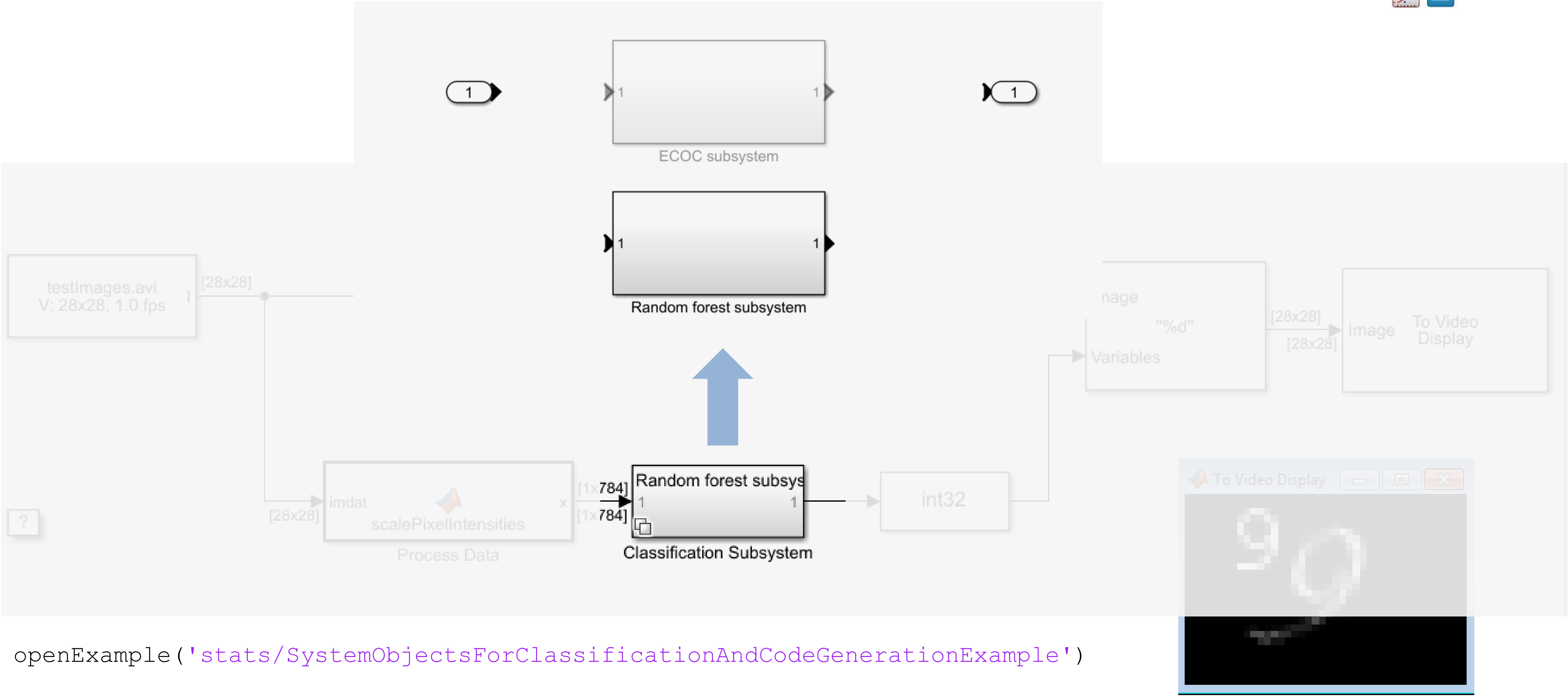
```
ensembleModel = 

  struct with fields:

                          predictFcn: @(x)exportableModel.predictFcn(predictorExtractionFcn(x))
              ClassificationEnsemble: [1×1 classreg.learning.classif.CompactClassificationEnsemble]
    HyperParameterOptimizationResult: [1×1 BayesianOptimization]
                               About: 'This struct is a trained model exported from Classification Learner R2020a.'
                        HowToPredict: 'To make predictions on a new predictor column matrix, X, use: ↲  yfit = c.predictFcn(X)
```

## Save Trained Model for Deployment



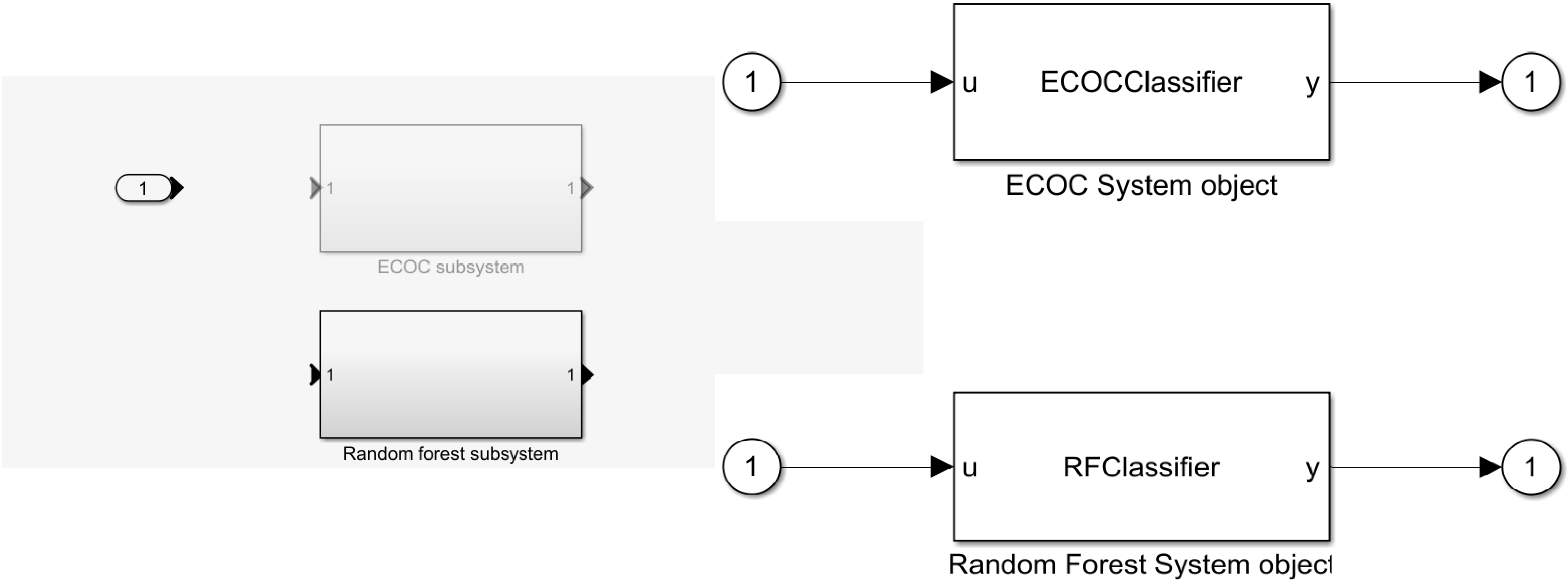
```
saveLearnerForCoder(ensembleModel.ClassificationEnsemble,'DigitImagesRF');
```

# Trained models can be used in Simulink

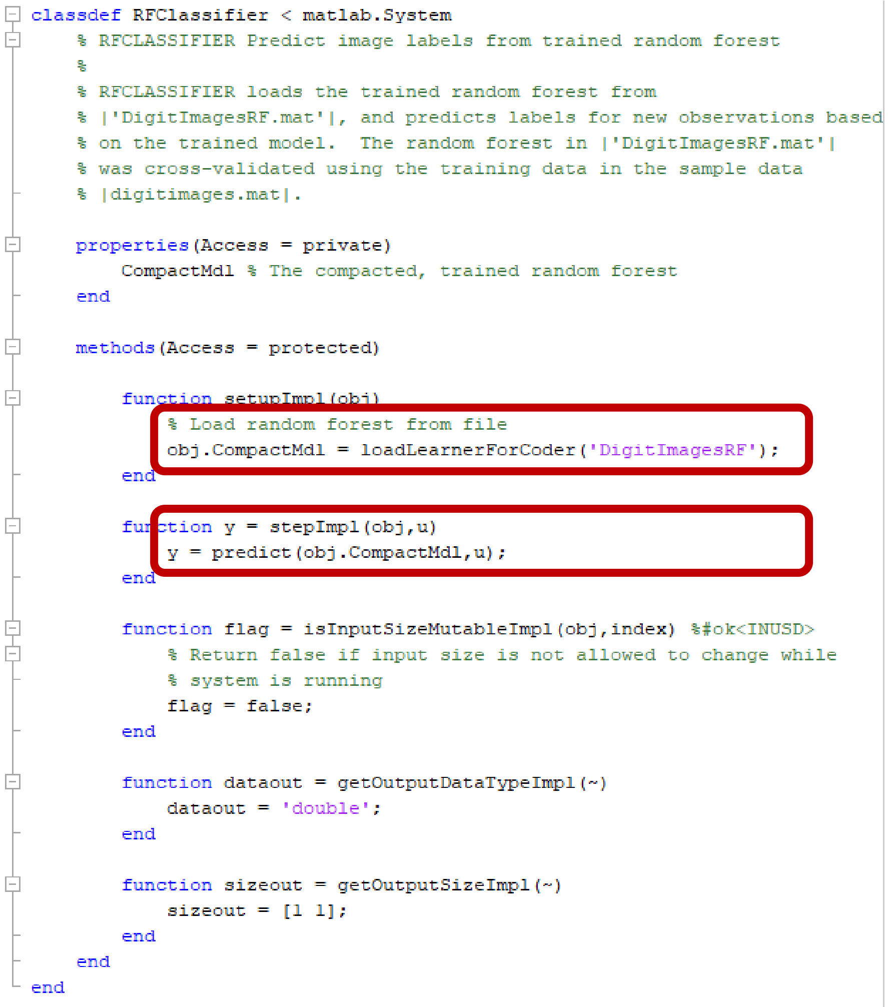


```
openExample('stats/SystemObjectsForClassificationAndCodeGenerationExample')
```

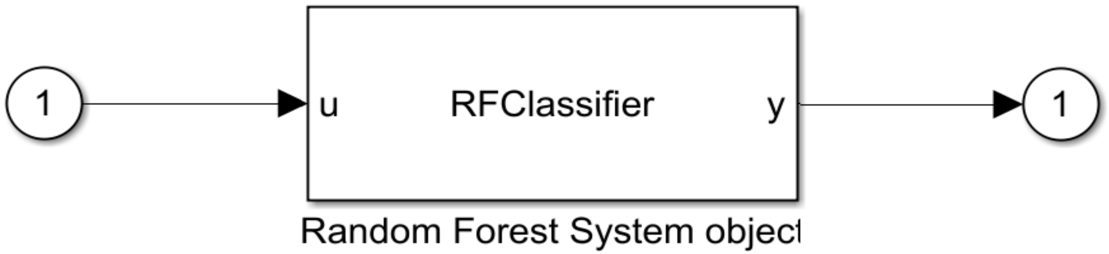
# Trained models can be used in Simulink



# Trained models can be used in Simulink via System Blocks



```
classdef RFClassifier < matlab.System
    % RFCLASSIFIER Predict image labels from trained random forest
    %
    % RFCLASSIFIER loads the trained random forest from
    % |'DigitImagesRF.mat'|, and predicts labels for new observations based
    % on the trained model.  The random forest in |'DigitImagesRF.mat'|
    % was cross-validated using the training data in the sample data
    % |digitimages.mat|.

    properties(Access = private)
        CompactMdl % The compacted, trained random forest
    end

    methods(Access = protected)

        function setupImpl(obj)
            % Load random forest from file
            obj.CompactMdl = loadLearnerForCoder('DigitImagesRF');
        end

        function y = stepImpl(obj,u)
            y = predict(obj.CompactMdl,u);
        end

        function flag = isInputSizeMutableImpl(obj,index) %#ok<INUSD>
            % Return false if input size is not allowed to change while
            % system is running
            flag = false;
        end

        function dataout = getOutputDataTypeImpl(~)
            dataout = 'double';
        end

        function sizeout = getOutputSizeImpl(~)
            sizeout = [1 1];
        end
    end
end
```

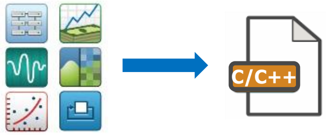


# Majority of machine learning models are supported for deployment

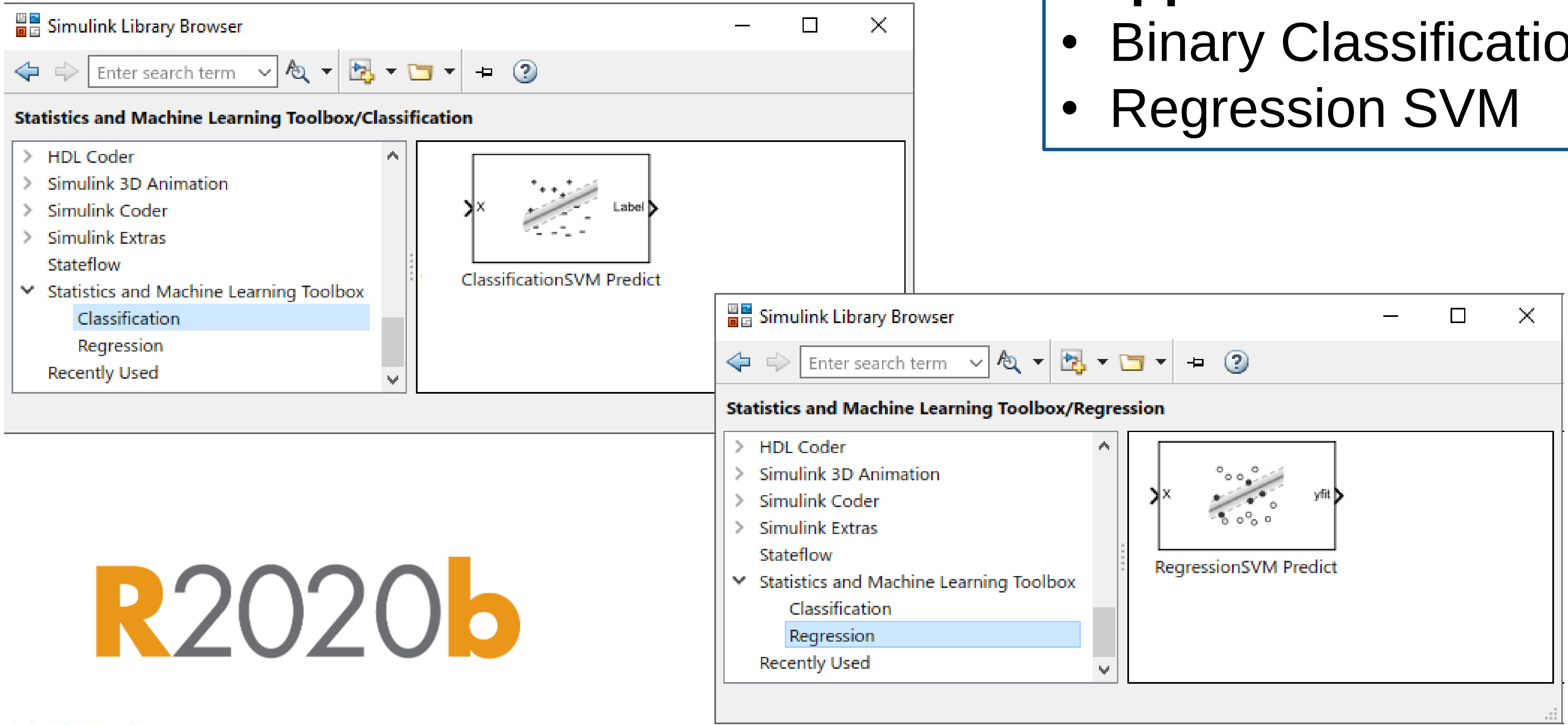
**Supported Models**

- Linear Classification
- SVM
- Decision trees and Random Forests
- Linear Discriminant Analysis
- k-Nearest Neighbor models
- Ensemble models
- Naïve Bayes models
- Gaussian Process
- Linear/Generalized Linear Regression

***Deploy machine learning models in MATLAB & Simulink***



# Native Simulink Library Blocks



Simulink Library Browser

Enter search term

Statistics and Machine Learning Toolbox/Classification

- HDL Coder
- Simulink 3D Animation
- Simulink Coder
- Simulink Extras
- Stateflow
- Statistics and Machine Learning Toolbox
  - Classification
  - Regression
- Recently Used

ClassificationSVM Predict

Simulink Library Browser

Enter search term

Statistics and Machine Learning Toolbox/Regression

- HDL Coder
- Simulink 3D Animation
- Simulink Coder
- Simulink Extras
- Stateflow
- Statistics and Machine Learning Toolbox
  - Classification
  - Regression
- Recently Used

RegressionSVM Predict

## Supported Models

- Binary Classification SVM
- Regression SVM

R2020b

# Majority of machine learning models are supported for deployment

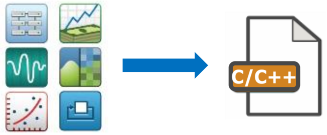
**Supported Models**

- Linear Classification
- SVM
- Decision trees and Random Forests
- Linear Discriminant Analysis
- k-Nearest Neighbor models
- Ensemble models
- Naïve Bayes models
- Gaussian Process
- Linear/Generalized Linear Regression

**Simulink**

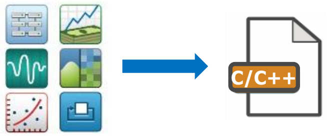
- Simulink Library Blocks
- MATLAB System Block
- MATLAB Function Block
- Stateflow

***Deploy machine learning models in MATLAB & Simulink***

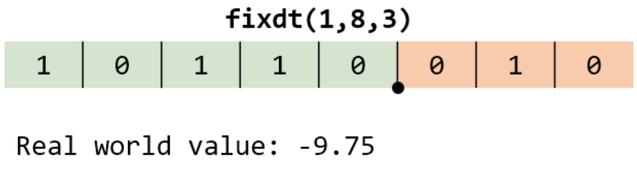


# Machine learning algorithms are supported for fixed-point workflows

***Deploy machine learning models in MATLAB & Simulink***



***Deploy fixed-point machine learning models***

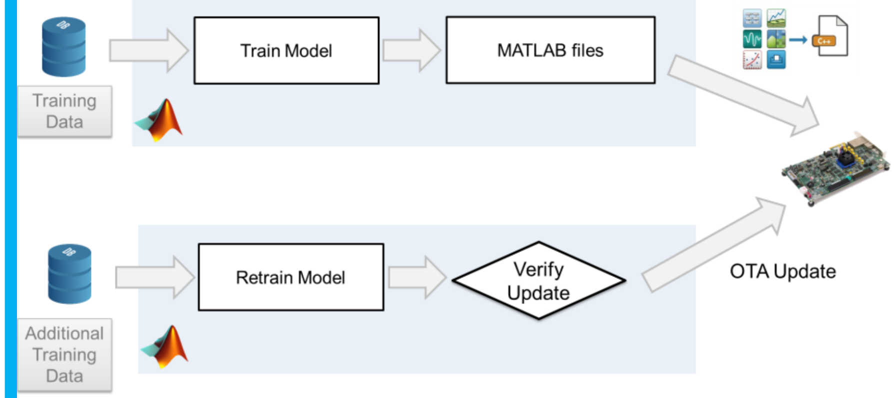


**fixdt(1,8,3)**

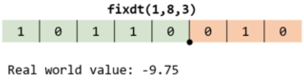
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

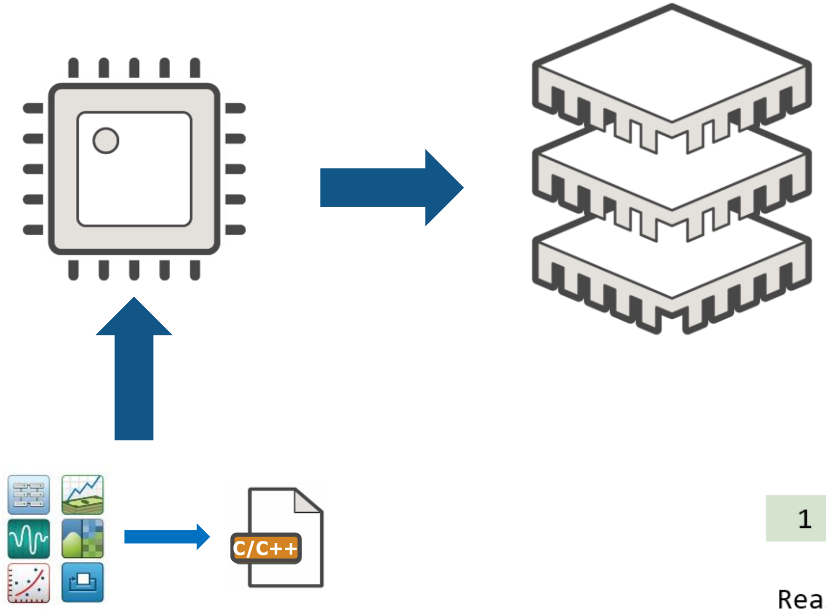
```
Real world value: -9.75
```

***In-place modification of deployed models***



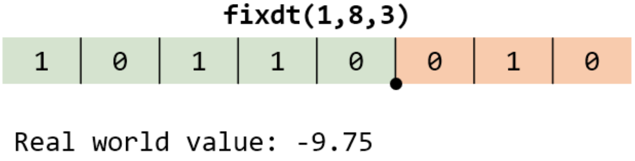
# Deploy fixed-point machine learning models





**Minimize energy consumption**

**Reduce cost**



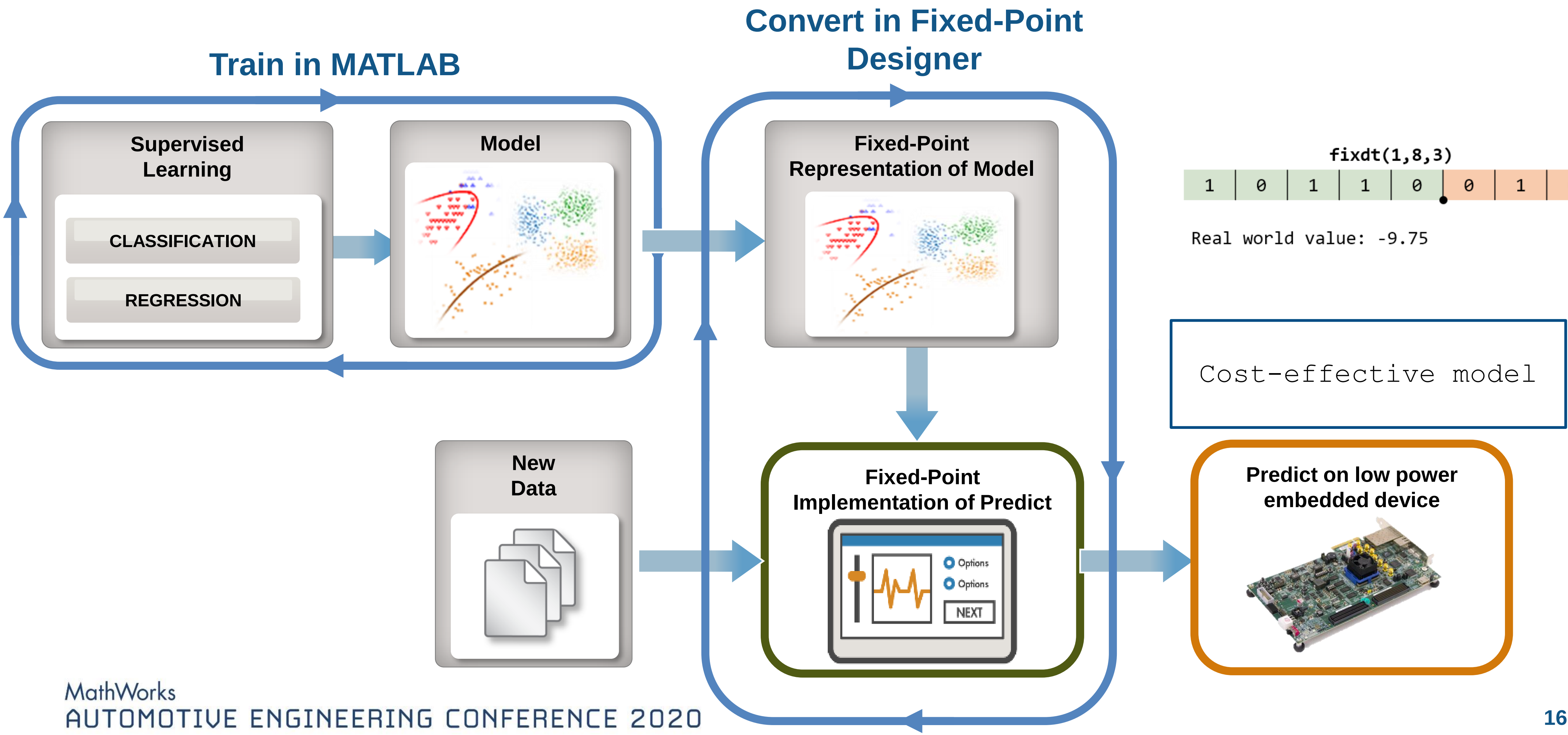
fixdt(1,8,3)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

Real world value: -9.75

# Fixed-point workflows allow deployment of models with small memory footprint

```
fixdt(1,8,3)
1 | 0 | 1 | 1 | 0 | 0 | 1 | 0
Real world value: -9.75
```

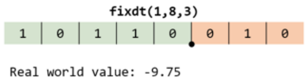


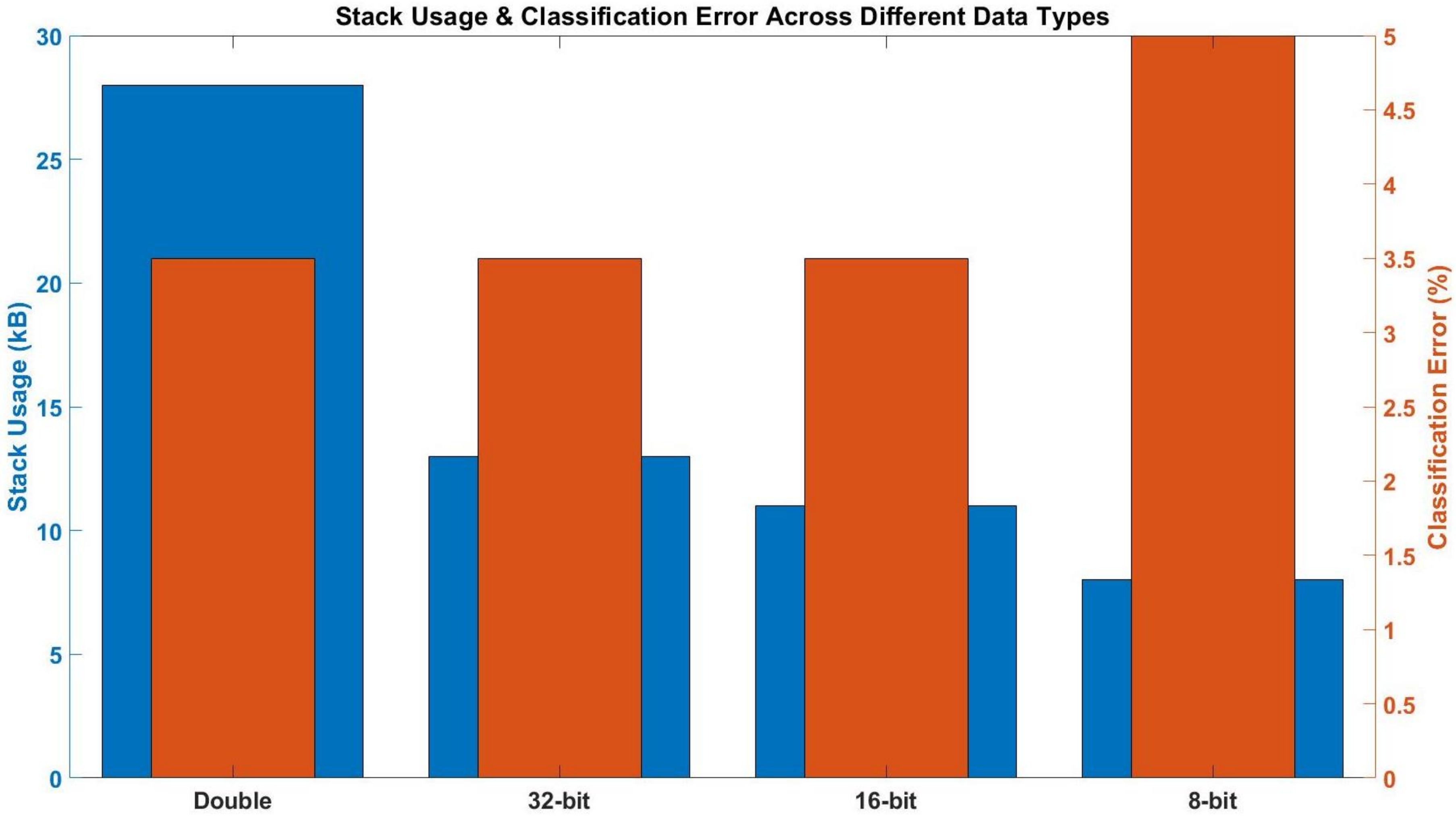
Train in MATLAB
Supervised Learning
CLASSIFICATION
REGRESSION
Model
Convert in Fixed-Point Designer
Fixed-Point Representation of Model
Fixed-Point Implementation of Predict
Options
Options
NEXT
New Data
Predict on low power embedded device

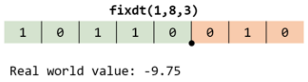
```
fixdt(1,8,3)
1 | 0 | 1 | 1 | 0 | 0 | 1 | 0
Real world value: -9.75
```

Cost-effective model

# Fixed-point conversion is a trade-off between resource usage optimization and accuracy






fixdt(1,8,3)

1 | 0 | 1 | 1 | 0 | 0 | 1 | 0

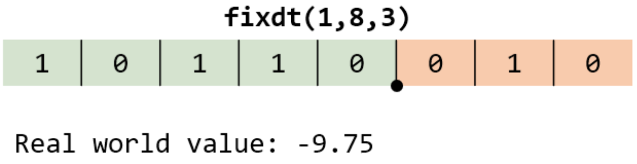
Real world value: -9.75

# Popular machine learning models are supported for fixed-point workflows

**Supported Models**

- Binary SVM
- Decision Trees
- Ensembles of Decision Trees

***Deploy fixed-point machine learning models***

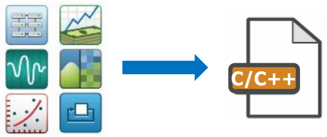

fixdt(1,8,3)

1 | 0 | 1 | 1 | 0 | 0 | 1 | 0

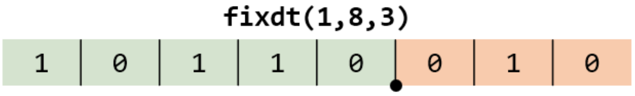
Real world value: -9.75

# Machine learning algorithms are supported for in-place modification workflows

## *Deploy machine learning models in MATLAB & Simulink*

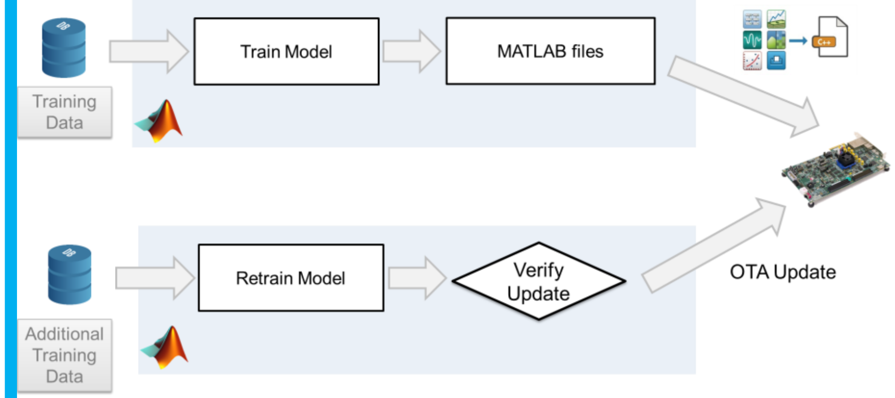


## *Deploy fixed-point machine learning models*

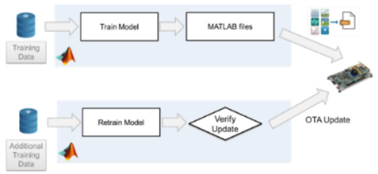


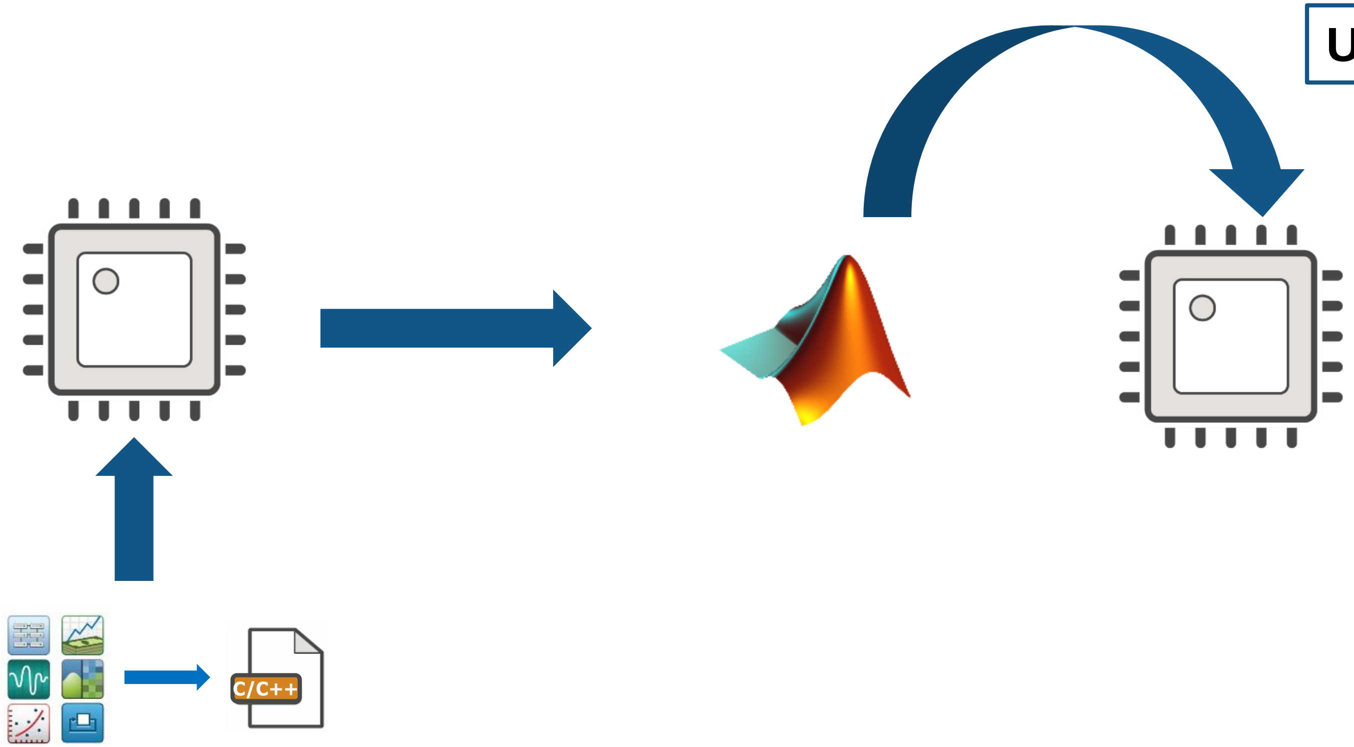
Real world value: -9.75

## *In-place modification of deployed models*



# In-place modification of deployed models

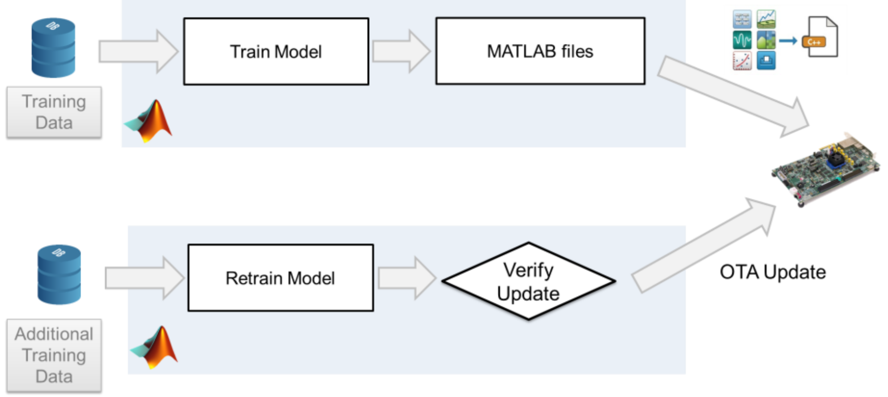




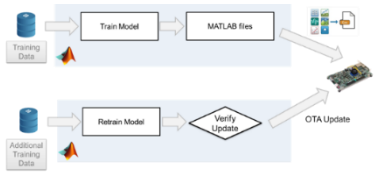
Update running model

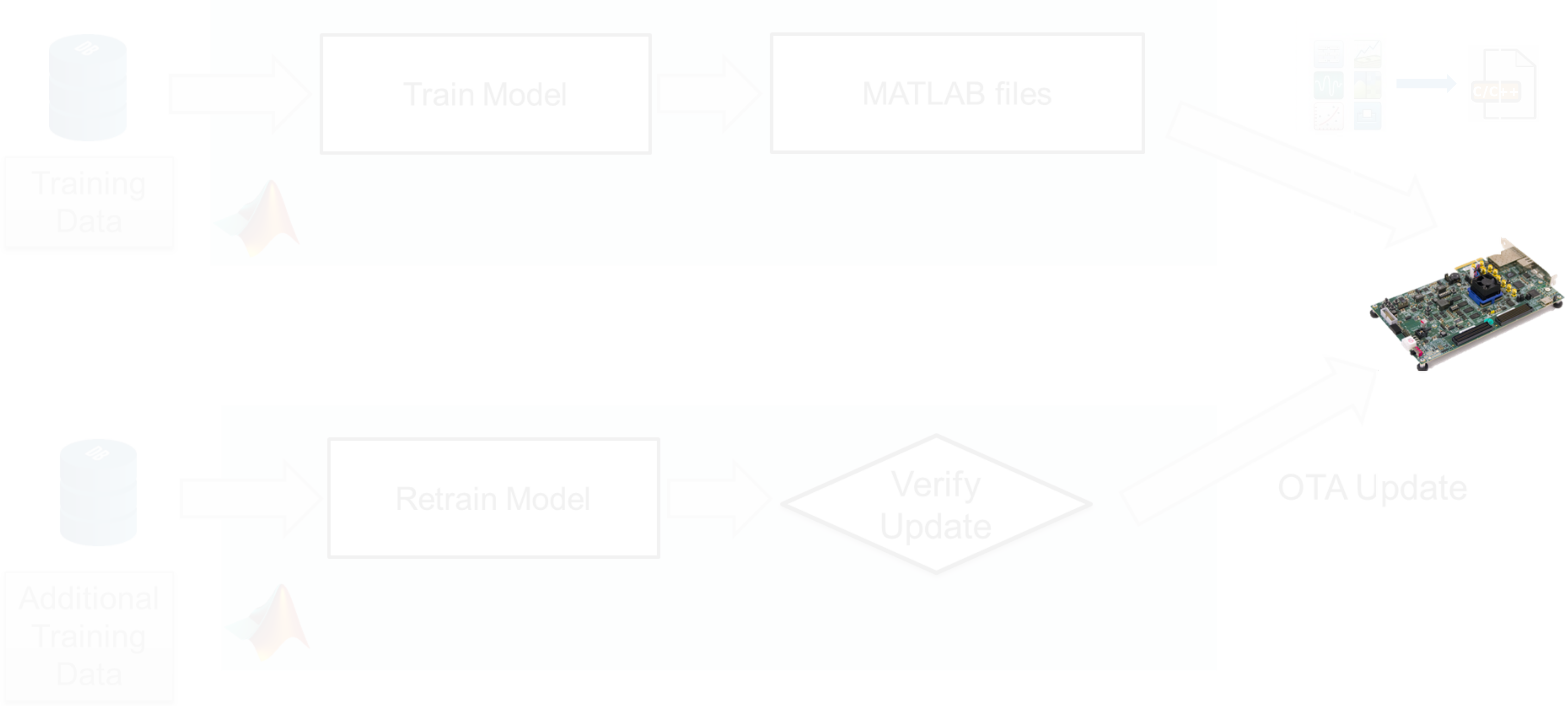
SIL/HIL Verification of models

OTA Update of models on remote vehicles

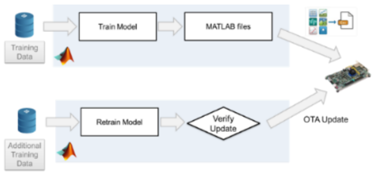


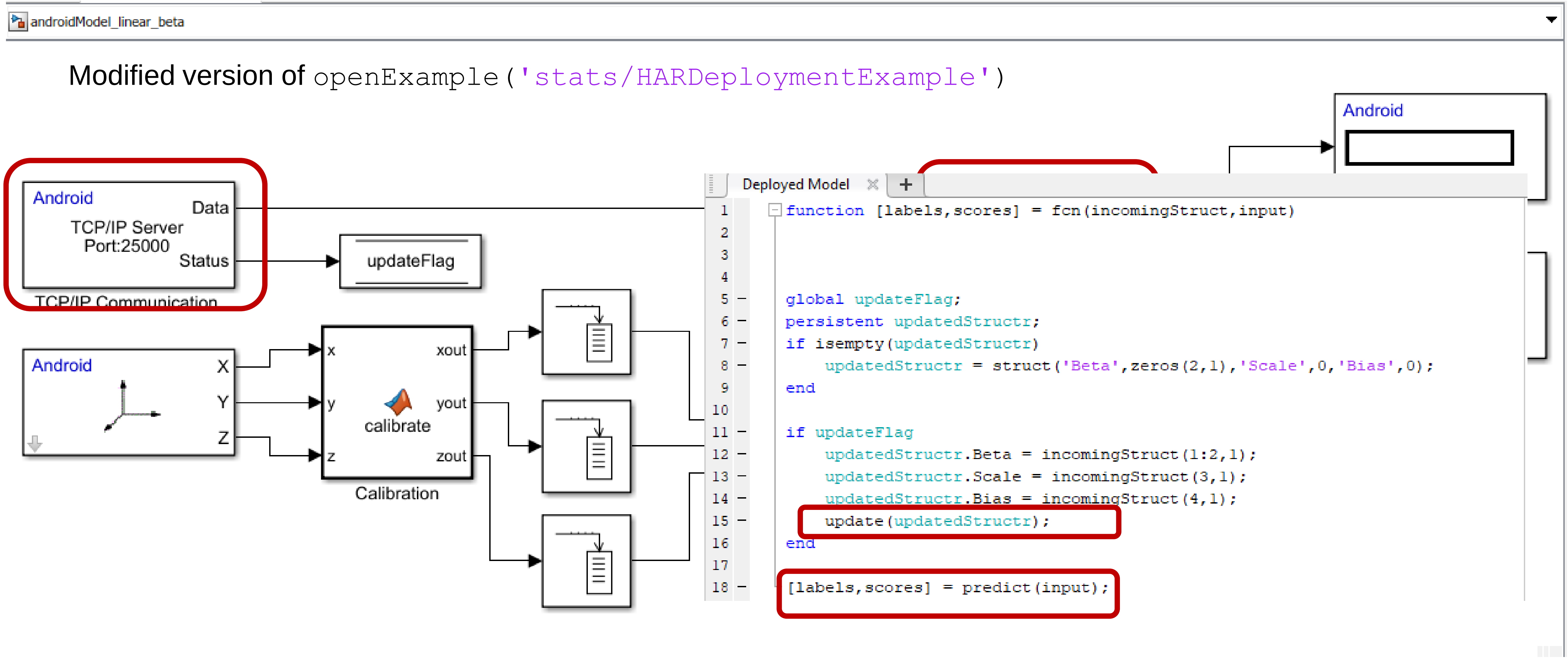
# In-place modification of deployed models allows model updates without code regeneration





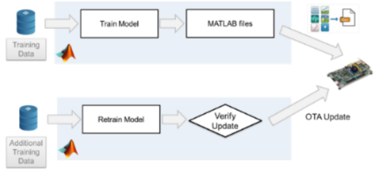
# In-place modification workflow is agnostic to communication method, supported in Simulink





Modified version of `openExample('stats/HARDeploymentExample')`

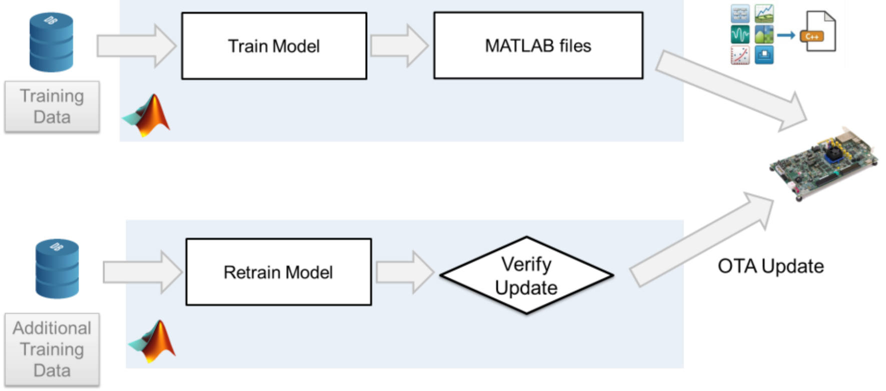
```
function [labels,scores] = fcn(incomingStruct,input)



global updateFlag;
persistent updatedStructr;
if isempty(updatedStructr)
    updatedStructr = struct('Beta',zeros(2,1),'Scale',0,'Bias',0);
end

if updateFlag
    updatedStructr.Beta = incomingStruct(1:2,1);
    updatedStructr.Scale = incomingStruct(3,1);
    updatedStructr.Bias = incomingStruct(4,1);
    update(updatedStructr);
end

[labels,scores] = predict(input);
```



# Popular machine learning models are supported for in-place modification workflows

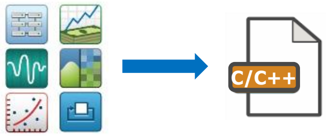
**Supported Models**
- SVM
- Linear Models
- Decision Trees

***In-place modification of deployed models***

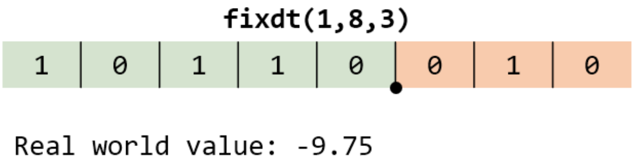


# Machine learning algorithms are supported for a variety of embedded systems workflows

***Deploy machine learning models in MATLAB & Simulink***



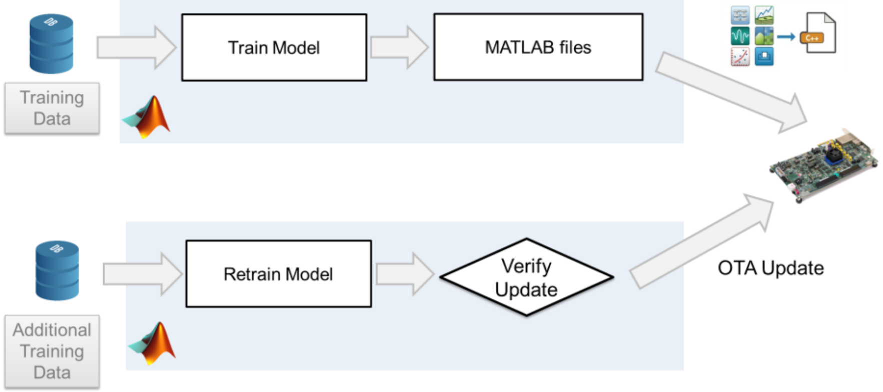
***Deploy fixed-point machine learning models***



```
fixdt(1,8,3)
 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0
```

```
Real world value: -9.75
```

***In-place modification of deployed models***



Training Data

Train Model

MATLAB files

Additional Training Data

Retrain Model

Verify Update

OTA Update

# Q & A

Which machine learning algorithms have you previously used in your projects?

| | | | |
|---|---|---|---|
| A | SVM | B | Decision Trees |
| C | Ensembles | D | Gaussian Process Models |
| E | KNN | F | Other |

If you have questions, please reach out

gatinc@mathworks.com

Are you already working on a project that involves deploying a machine learning model to an edge device?

| | | | |
|---|---|---|---|
| A | YES | B | NO |