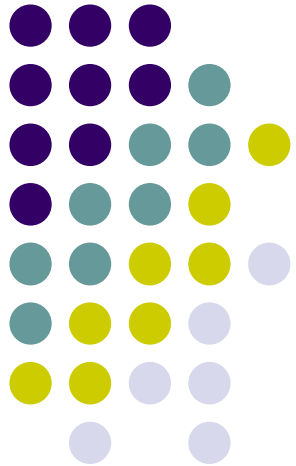
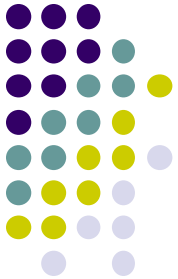
CS 528 Mobile and Ubiquitous Computing
MATLAB Machine Learner App

Emmanuel Agu

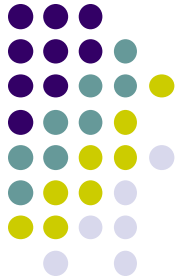




MathWorks

Agenda

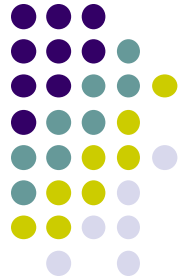
- Machine Learning
  - What is Machine Learning and why do we need it?
  - Common challenges in Machine Learning
- Example 1: Human activity learning using mobile phone data
  - Learning from sensor data
- Example 2: Real-time car identification using images
  - Learning from images
- Summary & Key Takeaways



MathWorks

Agenda

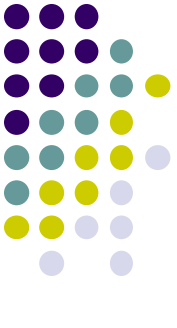
- Machine Learning
  - What is Machine Learning and why do we need it?
  - Common challenges in Machine Learning
- Example 1: Human activity learning using mobile phone data
  - Learning from sensor data
- Example 2: Real-time car identification using images
  - Learning from images
- Summary & Key Takeaways



MathWorks

Machine Learning is Everywhere

- Image Recognition
- Speech Recognition
- Stock Prediction
- Medical Diagnosis
- Data Analytics
- Robotics
- and more…



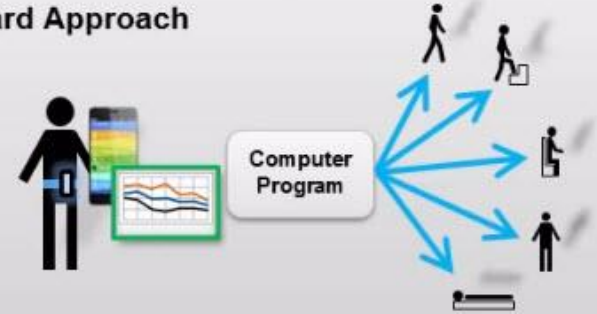
MathWorks

Machine Learning

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

Task: Human Activity Detection

Standard Approach



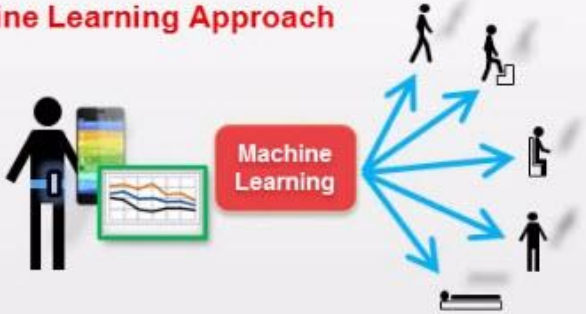
Hand Written Program

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

Formula or Equation

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

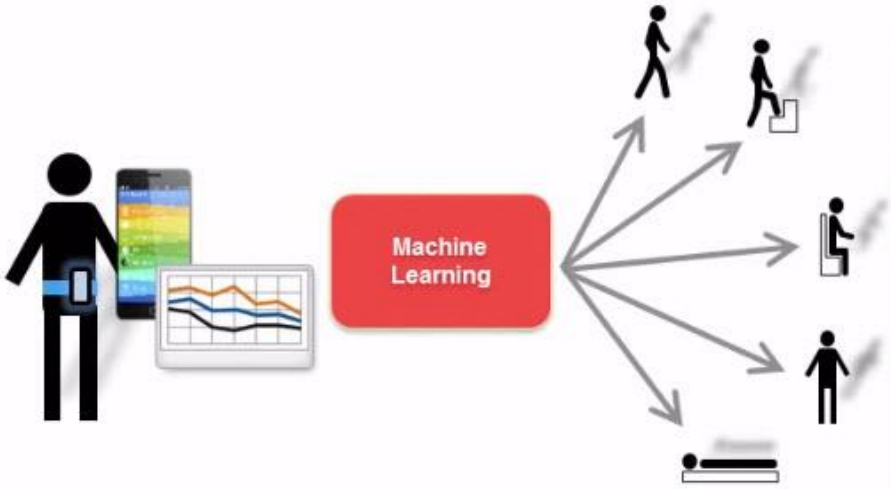
Machine Learning Approach

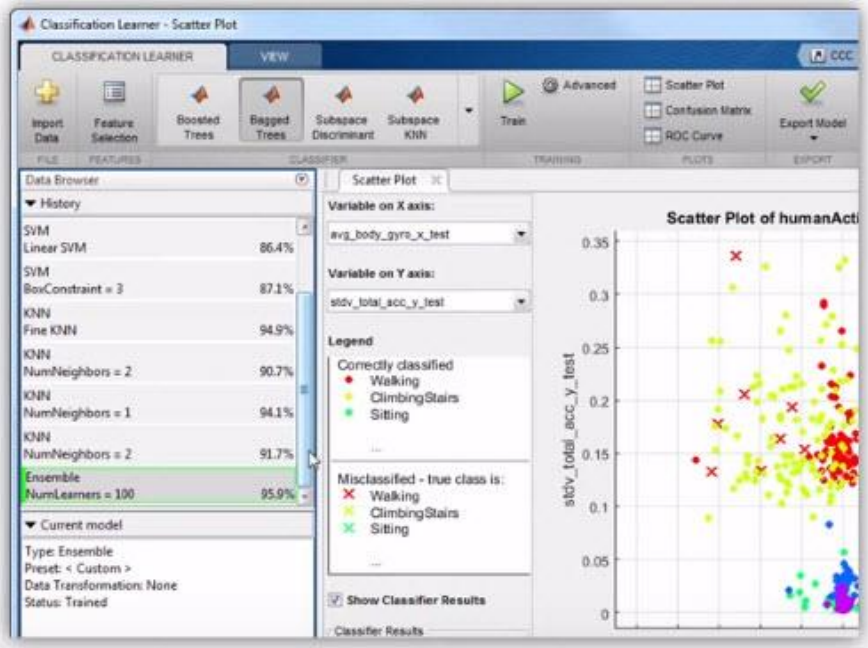


$model$: Inputs → Outputs

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

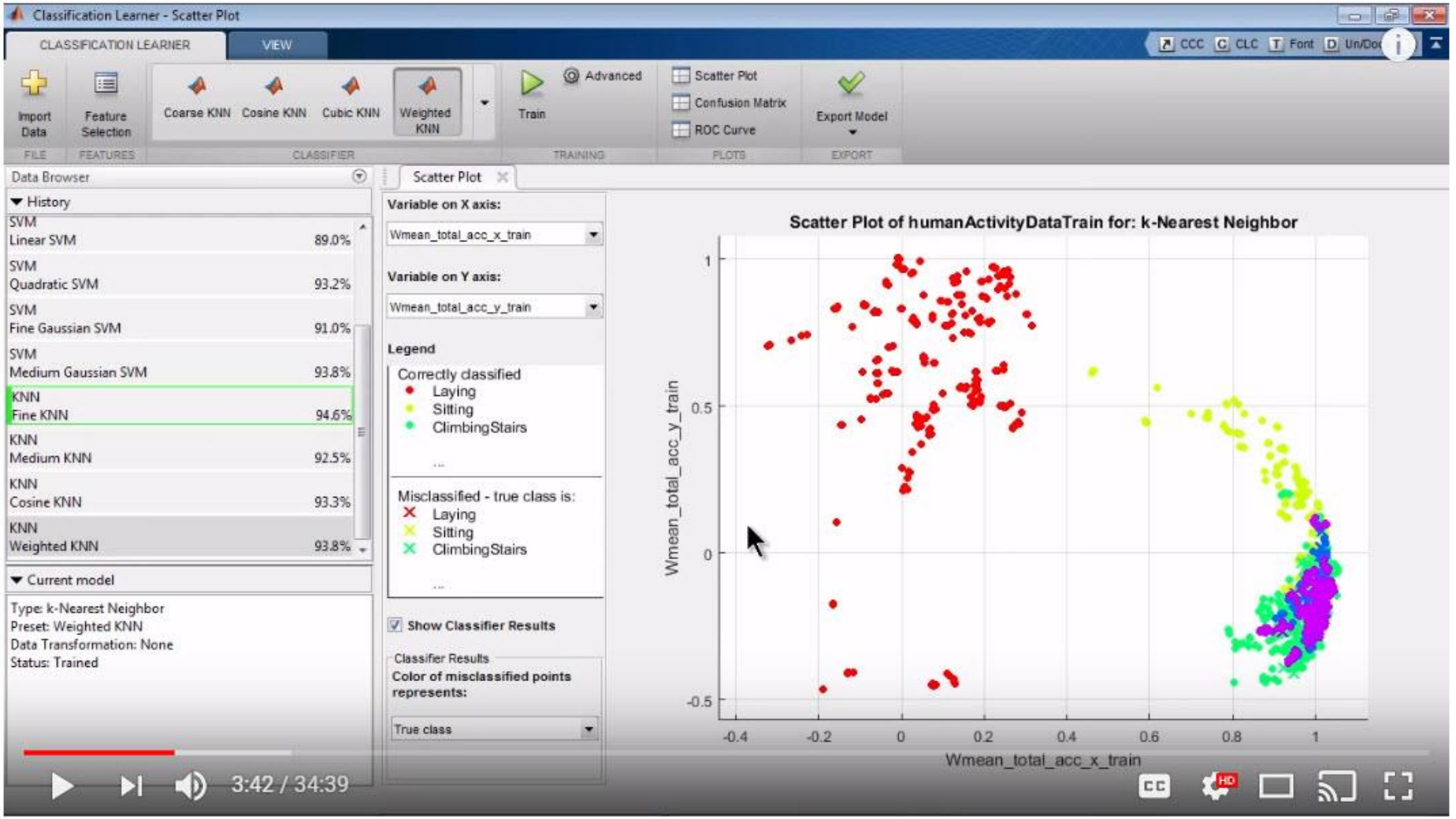
Example: Human Activity Learning Using Mobile Phone Data

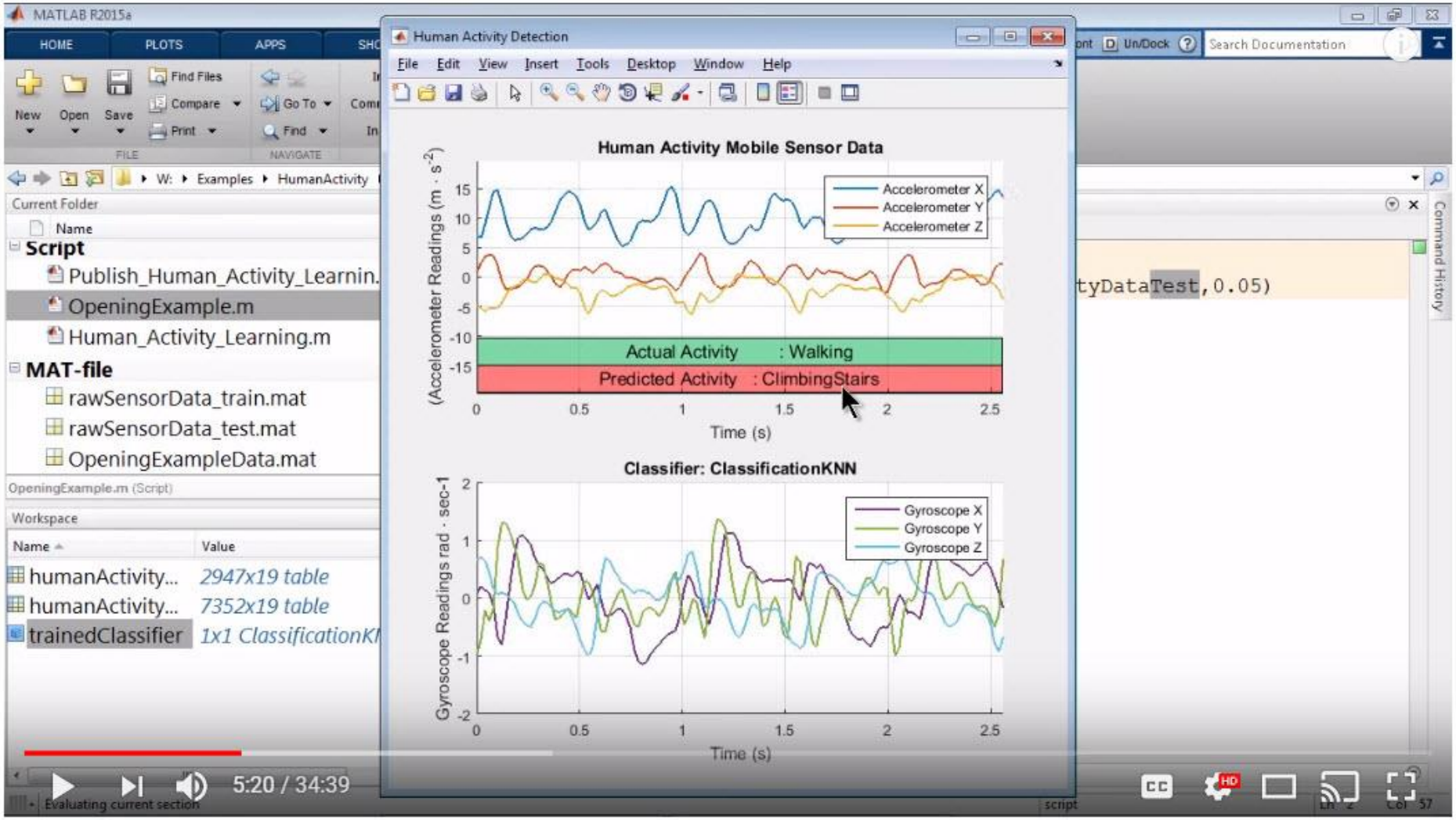


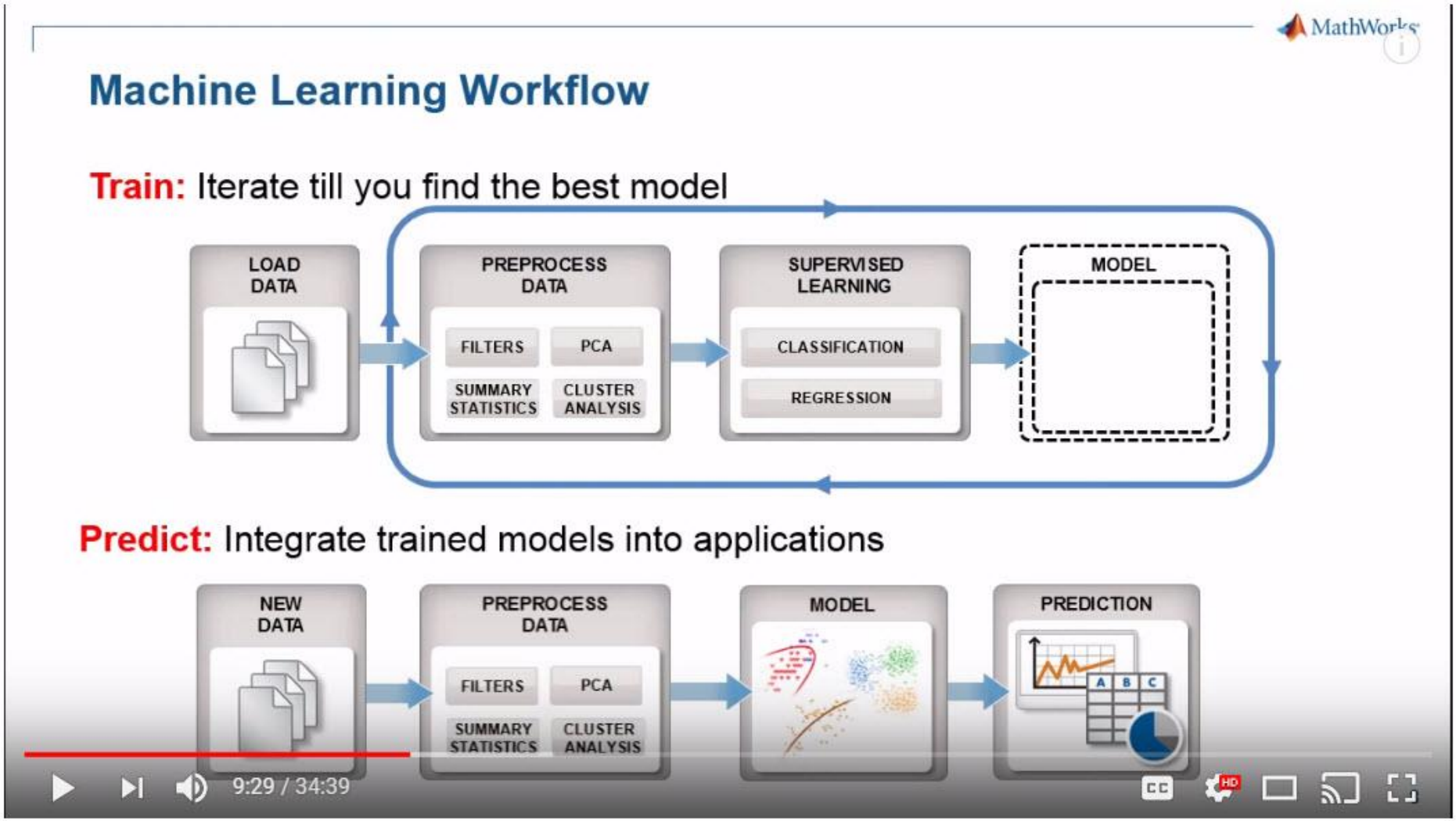


Data:

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


Classification Learner - Scatter Plot
CLASSIFICATION LEARNER
VIEW
Import Data
Feature Selection
Coarse KNN
Cosine KNN
Cubic KNN
Weighted KNN
Train
Advanced
Scatter Plot
Confusion Matrix
ROC Curve
Export Model
FILE
FEATURES
CLASSIFIER
TRAINING
PLOTS
EXPORT
Data Browser
History
SVM
Linear SVM 89.0%
SVM
Quadratic SVM 93.2%
SVM
Fine Gaussian SVM 91.0%
SVM
Medium Gaussian SVM 93.8%
KNN
Fine KNN 94.6%
KNN
Medium KNN 92.5%
KNN
Cosine KNN 93.3%
KNN
Weighted KNN 93.8%
Current model
Type: k-Nearest Neighbor
Preset: Weighted KNN
Data Transformation: None
Status: Trained
Scatter Plot
Variable on X axis:
Wmean_total_acc_x_train
Variable on Y axis:
Wmean_total_acc_y_train
Legend
Correctly classified
Laying
Sitting
ClimbingStairs
Misclassified - true class is:
Laying
Sitting
ClimbingStairs
Show Classifier Results
Classifier Results
Color of misclassified points represents:
True class
Scatter Plot of humanActivityDataTrain for: k-Nearest Neighbor
Wmean_total_acc_y_train
Wmean_total_acc_x_train
1
0.5
0
-0.5
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
3:42 / 34:39


MATLAB R2015a
HOME
PLOTS
APPS
New
Open
Save
Find Files
Compare
Print
Go To
Find
FILE
NAVIGATE
W: ▸ Examples ▸ HumanActivity
Current Folder
Name
Script
Publish_Human_Activity_Learnin.
OpeningExample.m
Human_Activity_Learning.m
MAT-file
rawSensorData_train.mat
rawSensorData_test.mat
OpeningExampleData.mat
OpeningExample.m (Script)
Workspace
Name
Value
humanActivity...
2947x19 table
humanActivity...
7352x19 table
trainedClassifier
1x1 ClassificationKN
Evaluating current section
5:20 / 34:39
Human Activity Detection
File Edit View Insert Tools Desktop Window Help
Human Activity Mobile Sensor Data
(Accelerometer Readings (m · s-2)
Accelerometer X
Accelerometer Y
Accelerometer Z
Actual Activity : Walking
Predicted Activity : ClimbingStairs
Time (s)
Classifier: ClassificationKNN
Gyroscope Readings rad · sec-1
Gyroscope X
Gyroscope Y
Gyroscope Z
Time (s)
Search Documentation
tyDataTest,0.05)
Command History
script

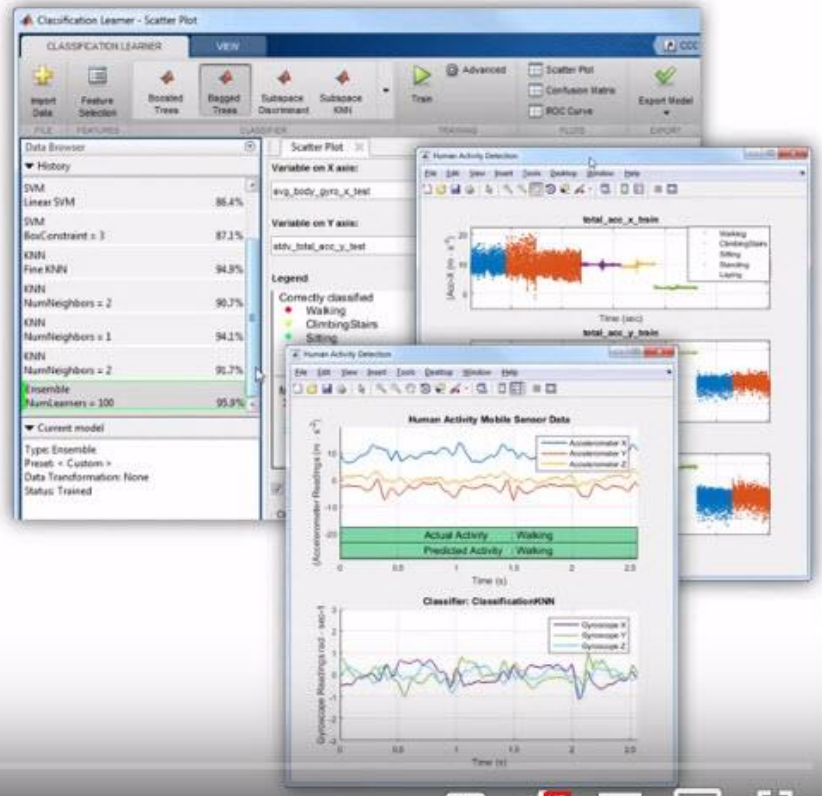

MathWorks
Machine Learning Workflow
Train: Iterate till you find the best model
LOAD DATA
PREPROCESS DATA
FILTERS
PCA
SUMMARY STATISTICS
CLUSTER ANALYSIS
SUPERVISED LEARNING
CLASSIFICATION
REGRESSION
MODEL
Predict: Integrate trained models into applications
NEW DATA
PREPROCESS DATA
FILTERS
PCA
SUMMARY STATISTICS
CLUSTER ANALYSIS
MODEL
PREDICTION
A B C
9:29 / 34:39
CC
HD

Example 1: Human Activity Learning Using Mobile Phone Data

Objective: Train a classifier to classify human activity from sensor data

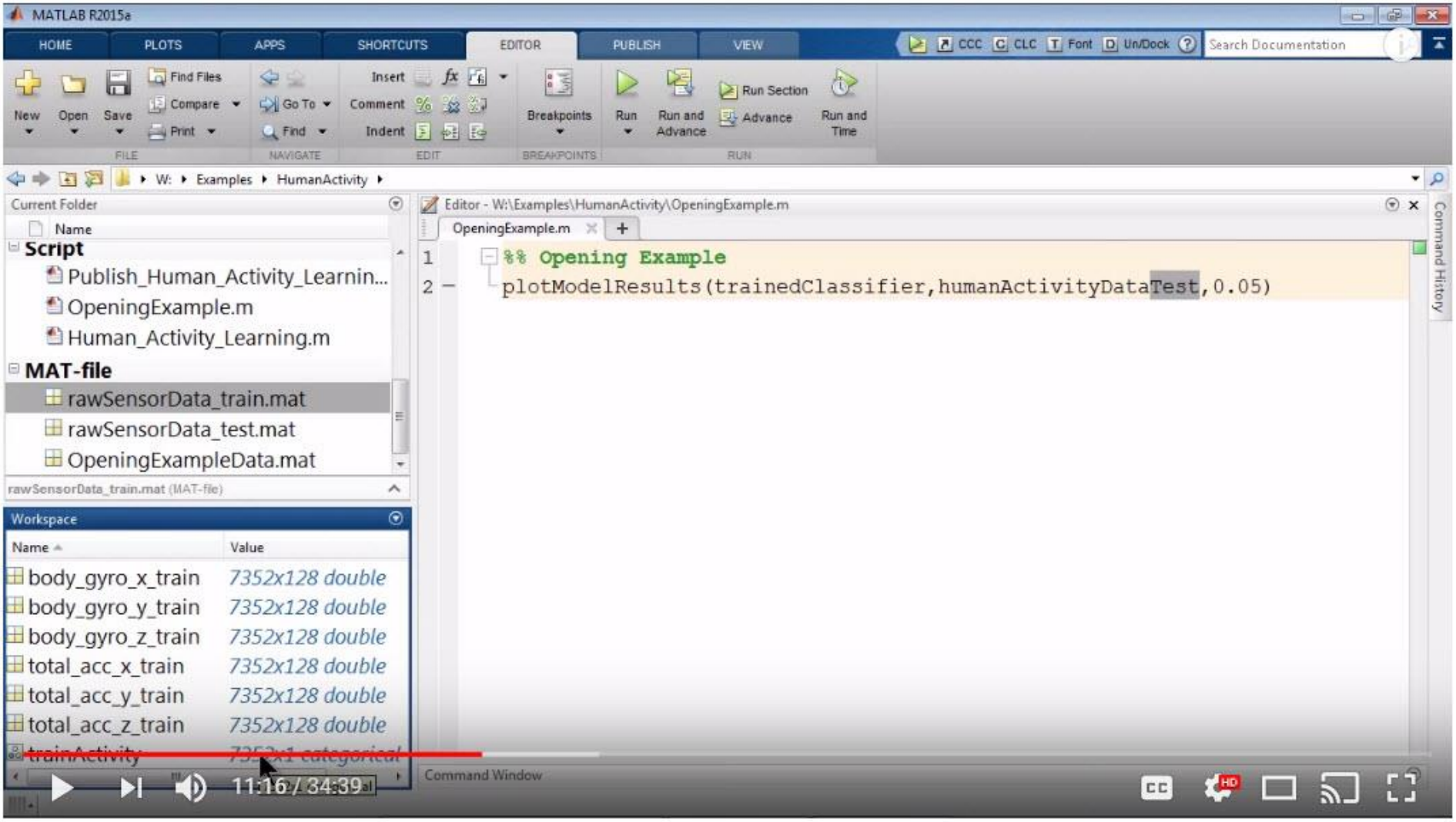
Data:

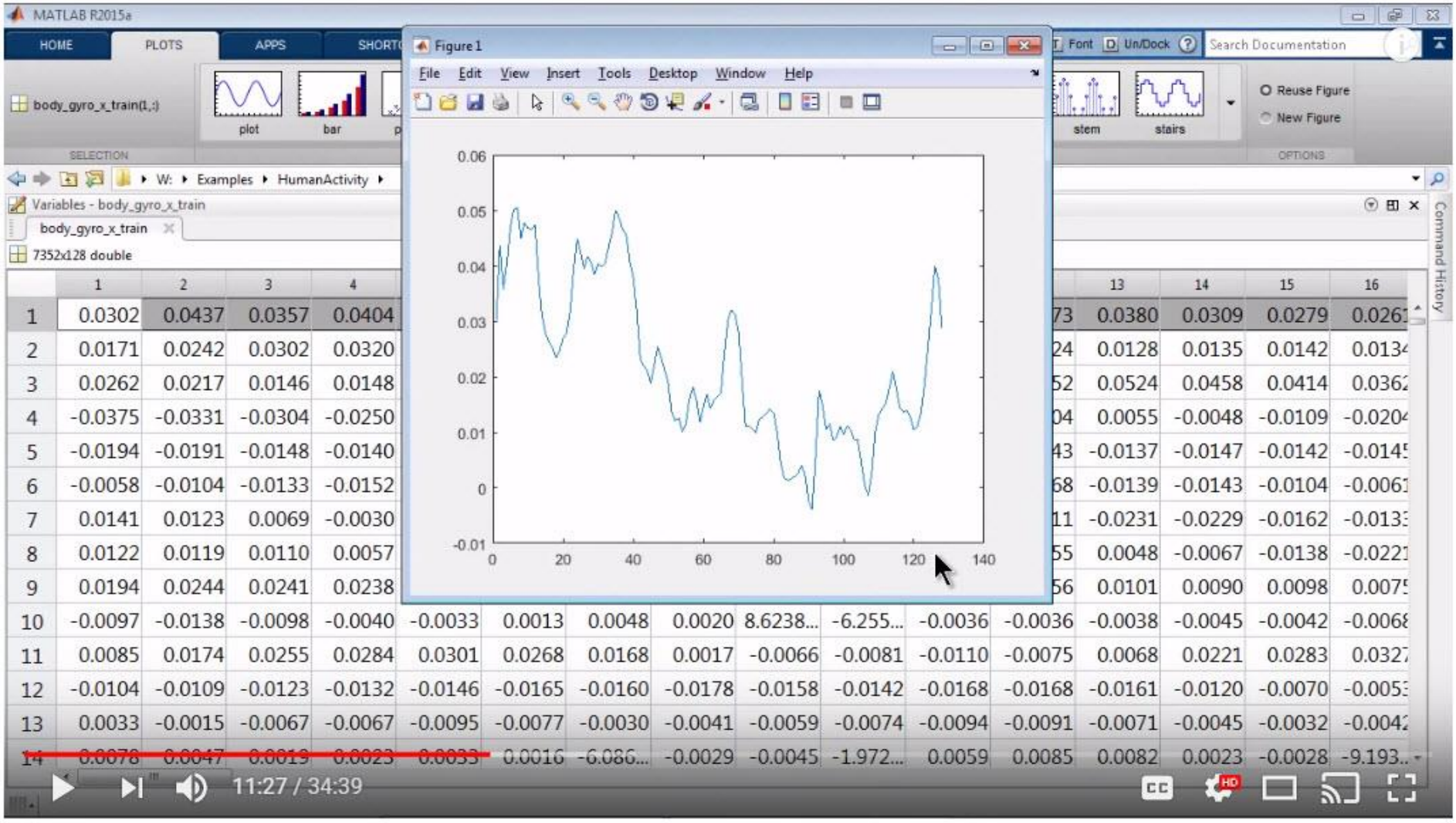
Predictors	3-axial Accelerometer and Gyroscope data 
Response	Activity: 

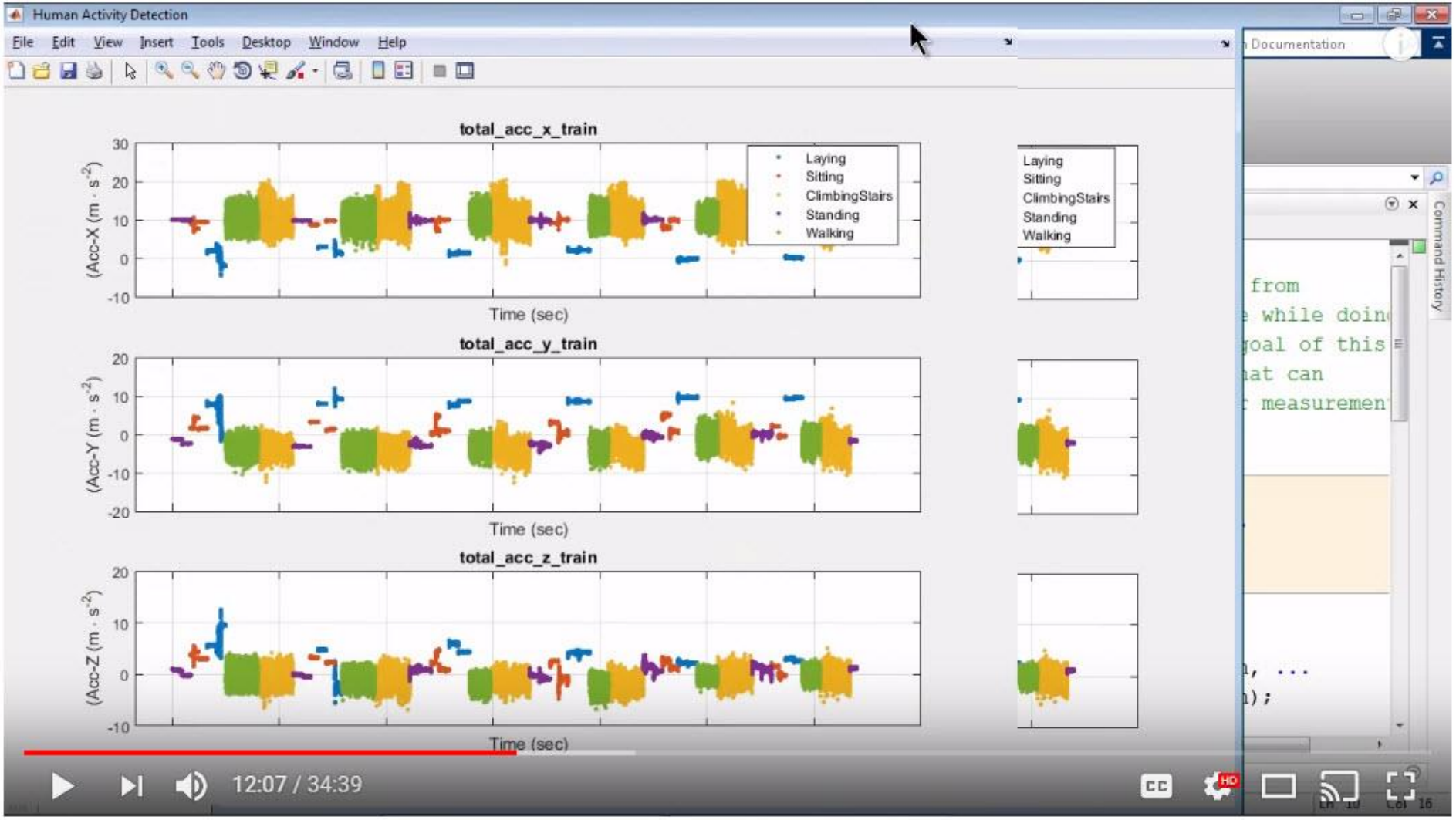


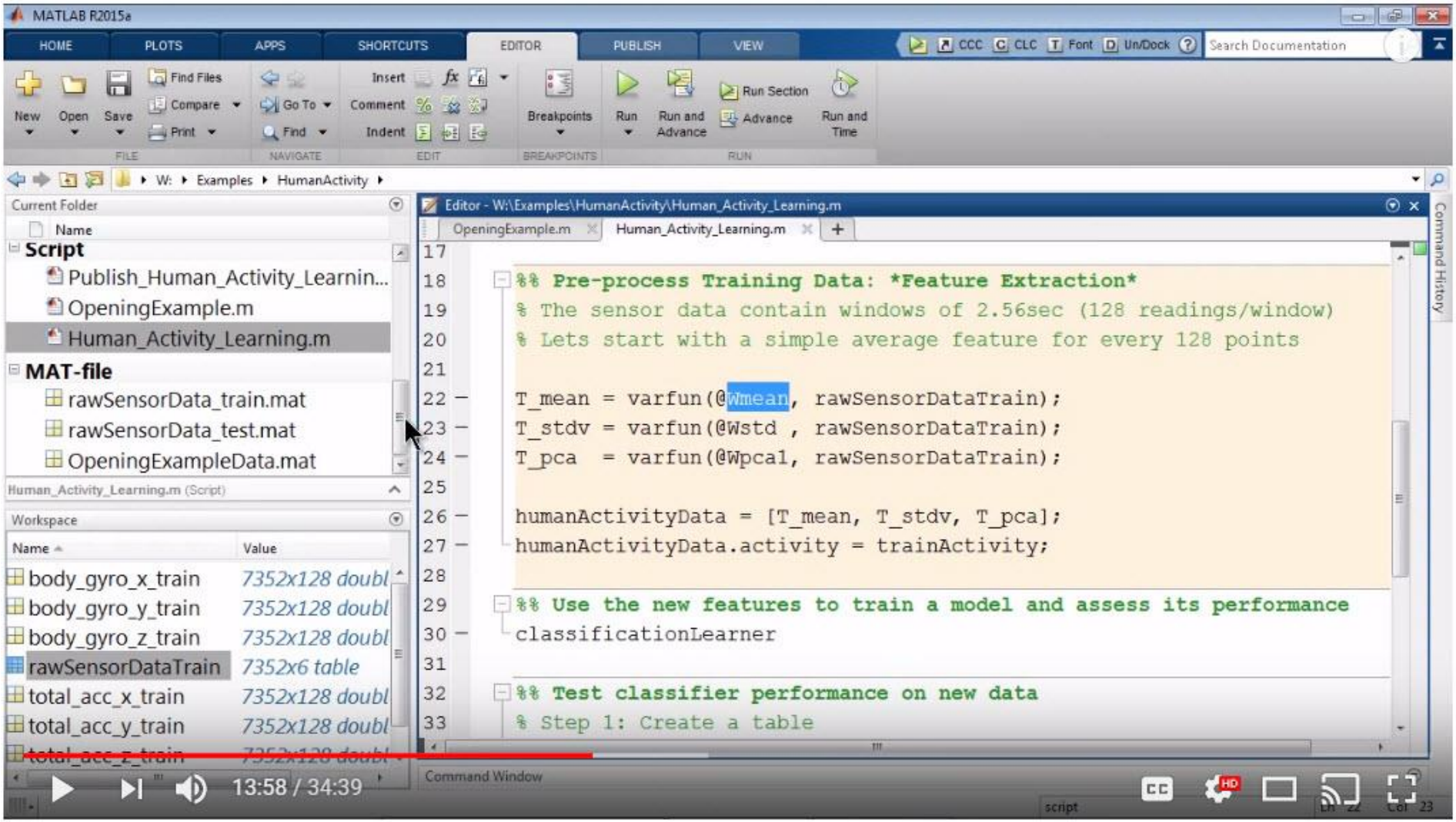
Approach:

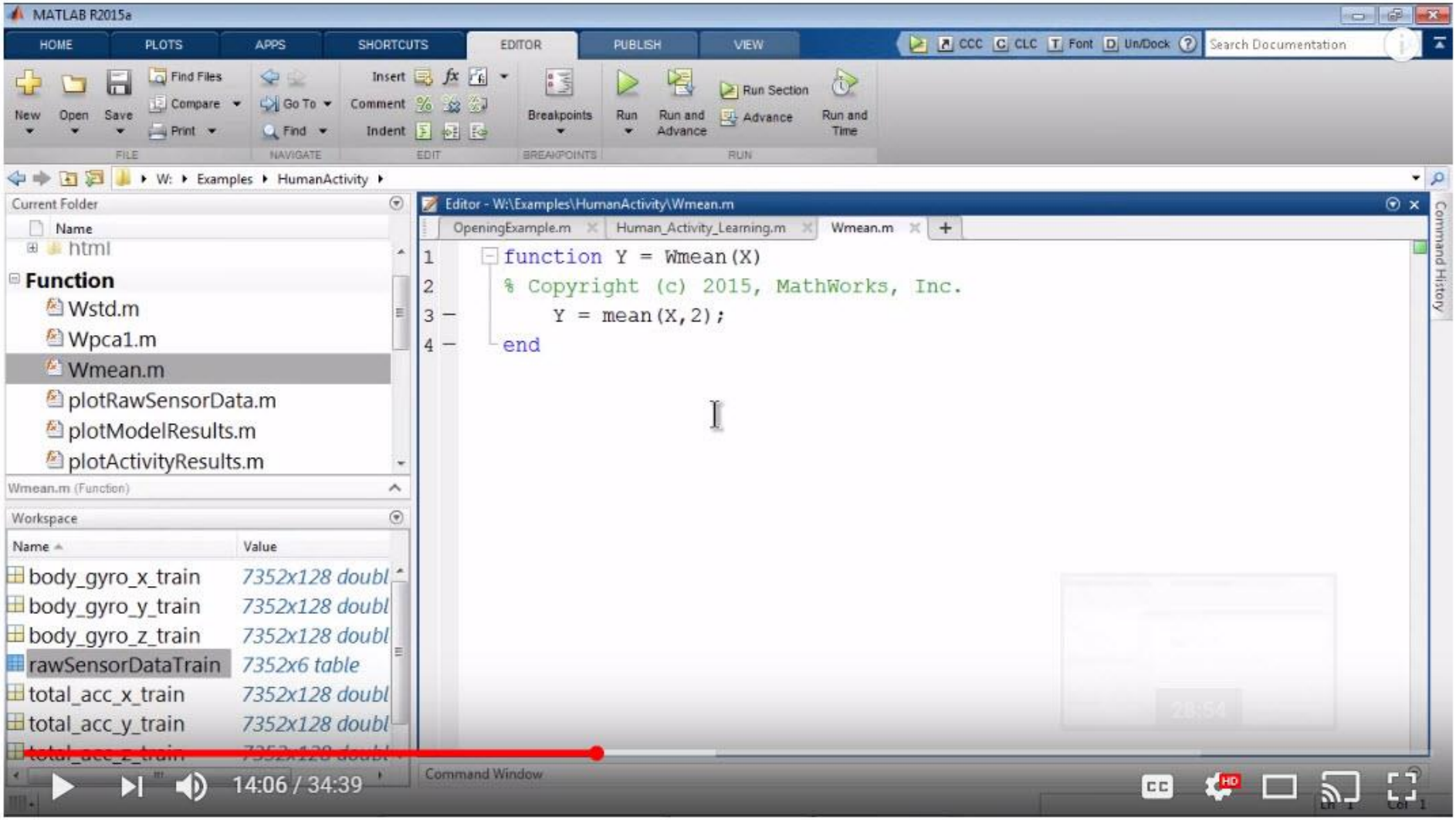
- Extract features from raw sensor signals
- Train and compare classifiers
- Test results on new sensor data

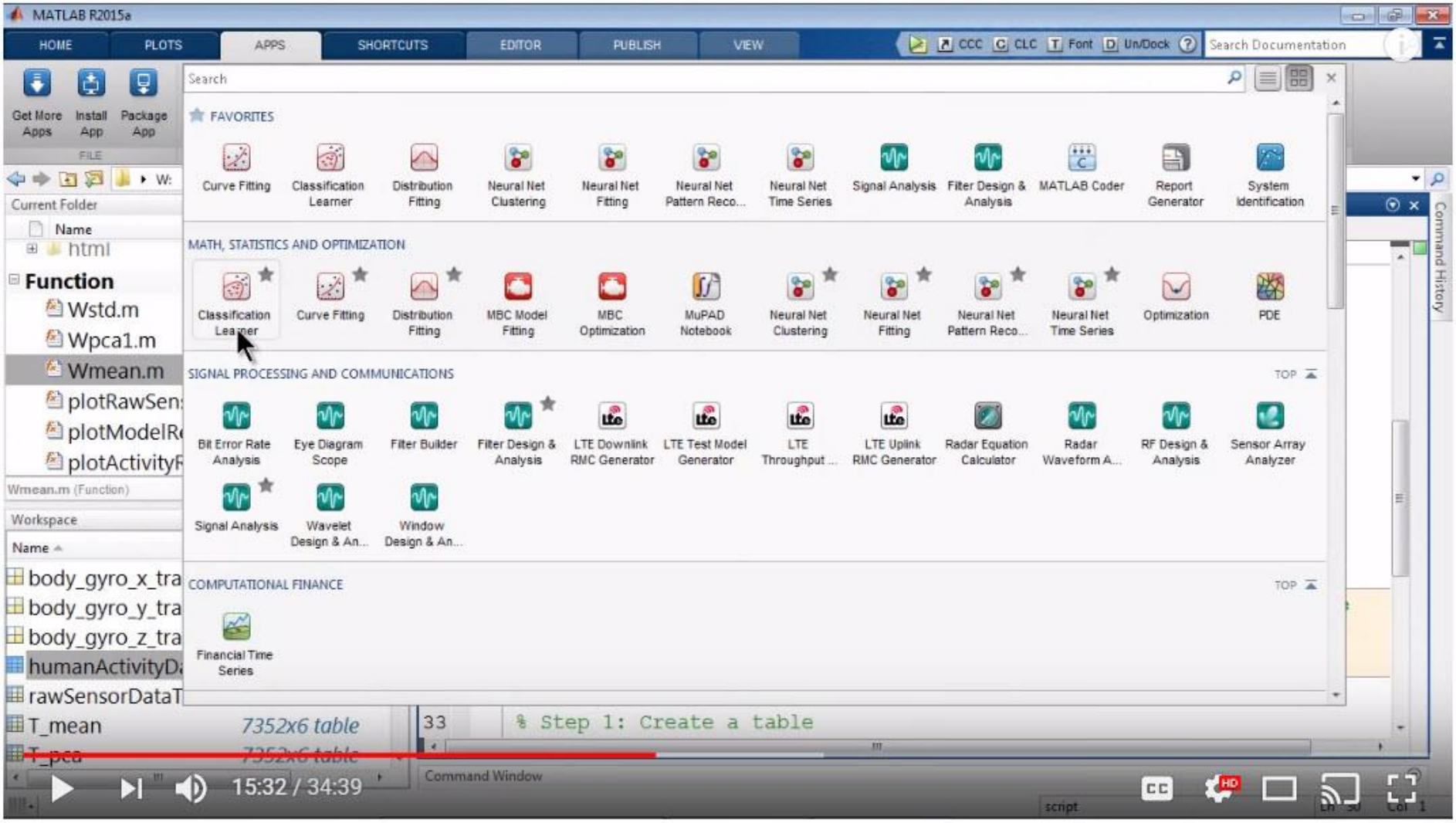

MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
W: ▸ Examples ▸ HumanActivity ▸
Current Folder
Name
Script
Publish_Human_Activity_Learnin...
OpeningExample.m
Human_Activity_Learning.m
MAT-file
rawSensorData_train.mat
rawSensorData_test.mat
OpeningExampleData.mat
rawSensorData_train.mat (MAT-file)
Workspace
Name
Value
body_gyro_x_train 7352x128 double
body_gyro_y_train 7352x128 double
body_gyro_z_train 7352x128 double
total_acc_x_train 7352x128 double
total_acc_y_train 7352x128 double
total_acc_z_train 7352x128 double
trainActivity
Editor - W:\Examples\HumanActivity\OpeningExample.m
OpeningExample.m
1 %% Opening Example
2 plotModelResults(trainedClassifier,humanActivityDataTest,0.05)
Command History
Command Window
11:16 / 34:39

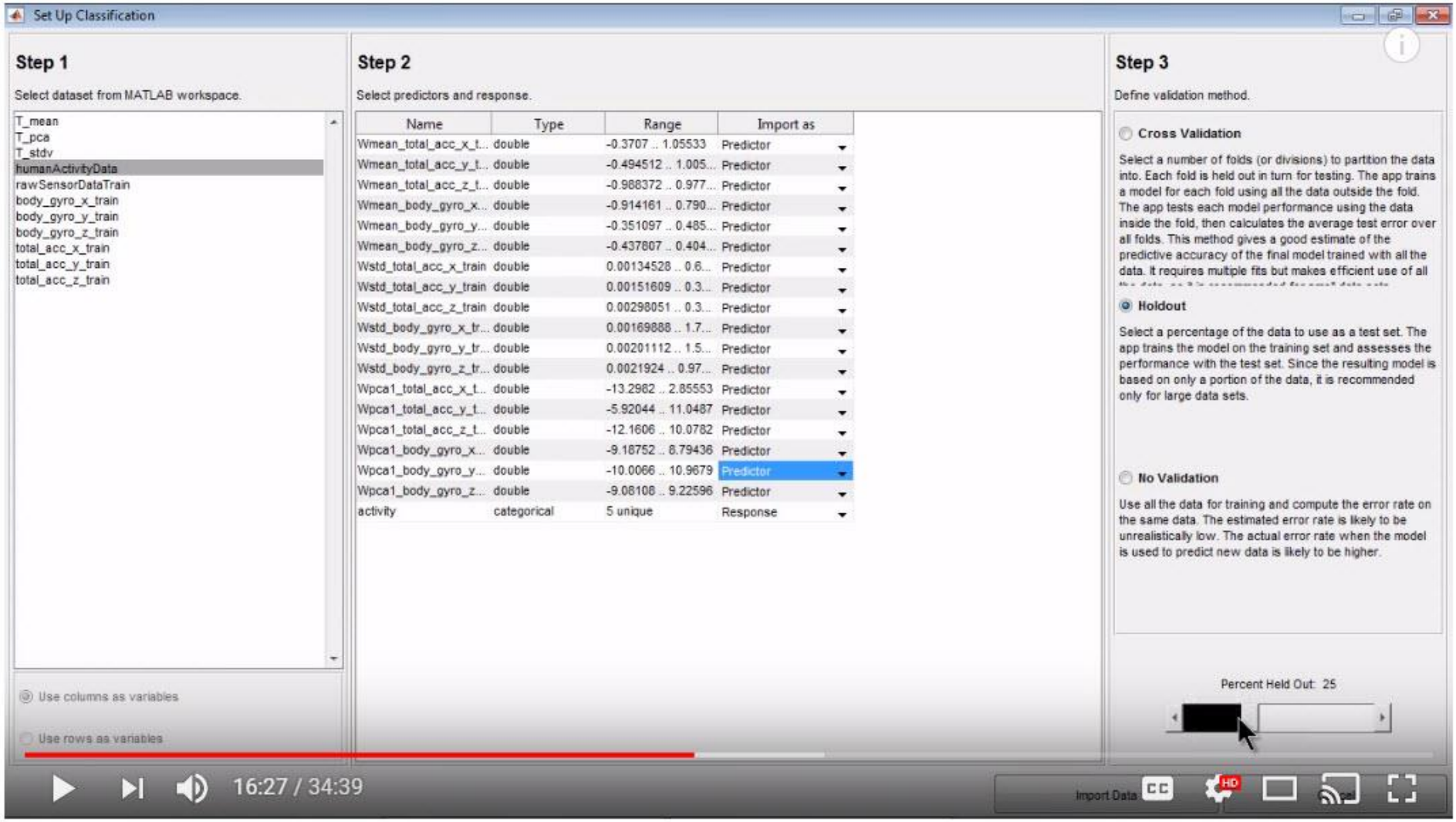

MATLAB R2015a
HOME
PLOTS
APPS
Figure 1
File Edit View Insert Tools Desktop Window Help
plot
bar
stem
stairs
Reuse Figure
New Figure
W: ▸ Examples ▸ HumanActivity ▸
Variables - body_gyro_x_train
body_gyro_x_train
7352x128 double
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
0 20 40 60 80 100 120 140
11:27 / 34:39

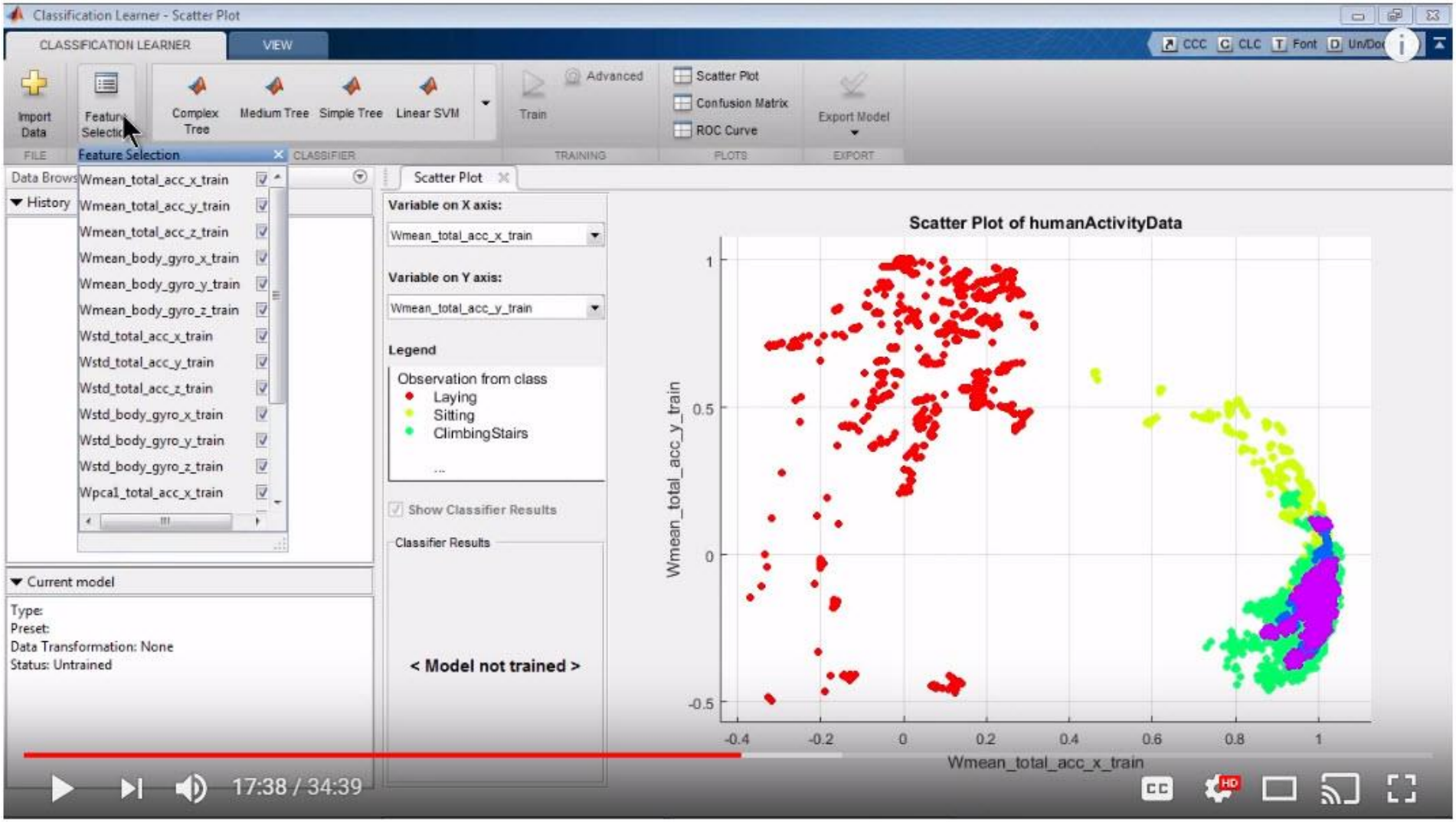

Human Activity Detection
File Edit View Insert Tools Desktop Window Help
Documentation
total_acc_x_train
(Acc-X (m · s-2))
Time (sec)
Laying
Sitting
ClimbingStairs
Standing
Walking
total_acc_y_train
(Acc-Y (m · s-2))
Time (sec)
total_acc_z_train
(Acc-Z (m · s-2))
Time (sec)
Command History
12:07 / 34:39

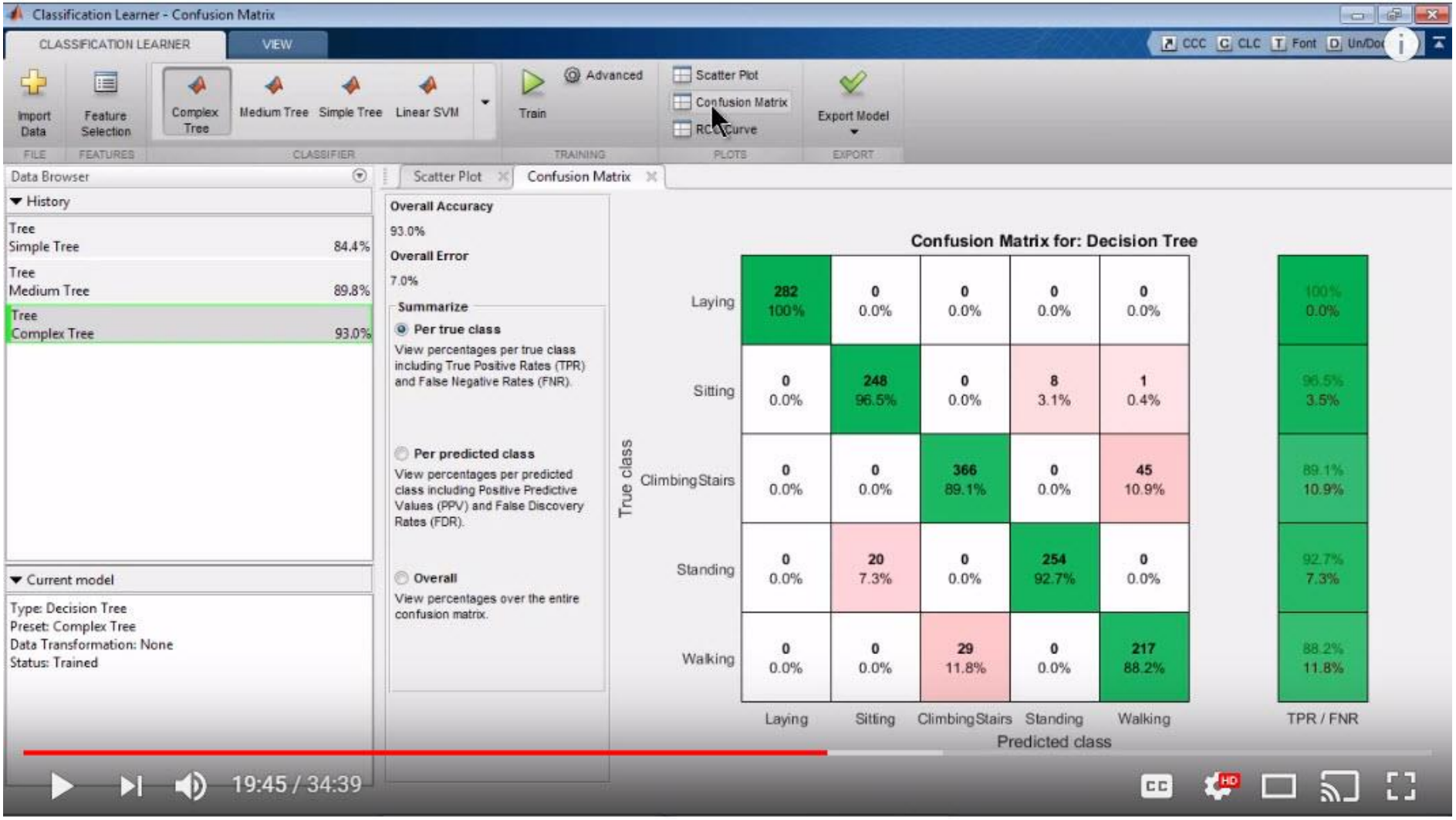

MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
W: ▸ Examples ▸ HumanActivity
Current Folder
Name
Script
Publish_Human_Activity_Learnin...
OpeningExample.m
Human_Activity_Learning.m
MAT-file
rawSensorData_train.mat
rawSensorData_test.mat
OpeningExampleData.mat
Human_Activity_Learning.m (Script)
Workspace
Name
Value
body_gyro_x_train 7352x128 doubl
body_gyro_y_train 7352x128 doubl
body_gyro_z_train 7352x128 doubl
rawSensorDataTrain 7352x6 table
total_acc_x_train 7352x128 doubl
total_acc_y_train 7352x128 doubl
Editor - W:\Examples\HumanActivity\Human_Activity_Learning.m
OpeningExample.m
Human_Activity_Learning.m
17
18 %% Pre-process Training Data: *Feature Extraction*
19 % The sensor data contain windows of 2.56sec (128 readings/window)
20 % Lets start with a simple average feature for every 128 points
21
22 T_mean = varfun(@Wmean, rawSensorDataTrain);
23 T_stdv = varfun(@Wstd , rawSensorDataTrain);
24 T_pca = varfun(@Wpca1, rawSensorDataTrain);
25
26 humanActivityData = [T_mean, T_stdv, T_pca];
27 humanActivityData.activity = trainActivity;
28
29 %% Use the new features to train a model and assess its performance
30 classificationLearner
31
32 %% Test classifier performance on new data
33 % Step 1: Create a table
Command Window
Command History
13:58 / 34:39
script

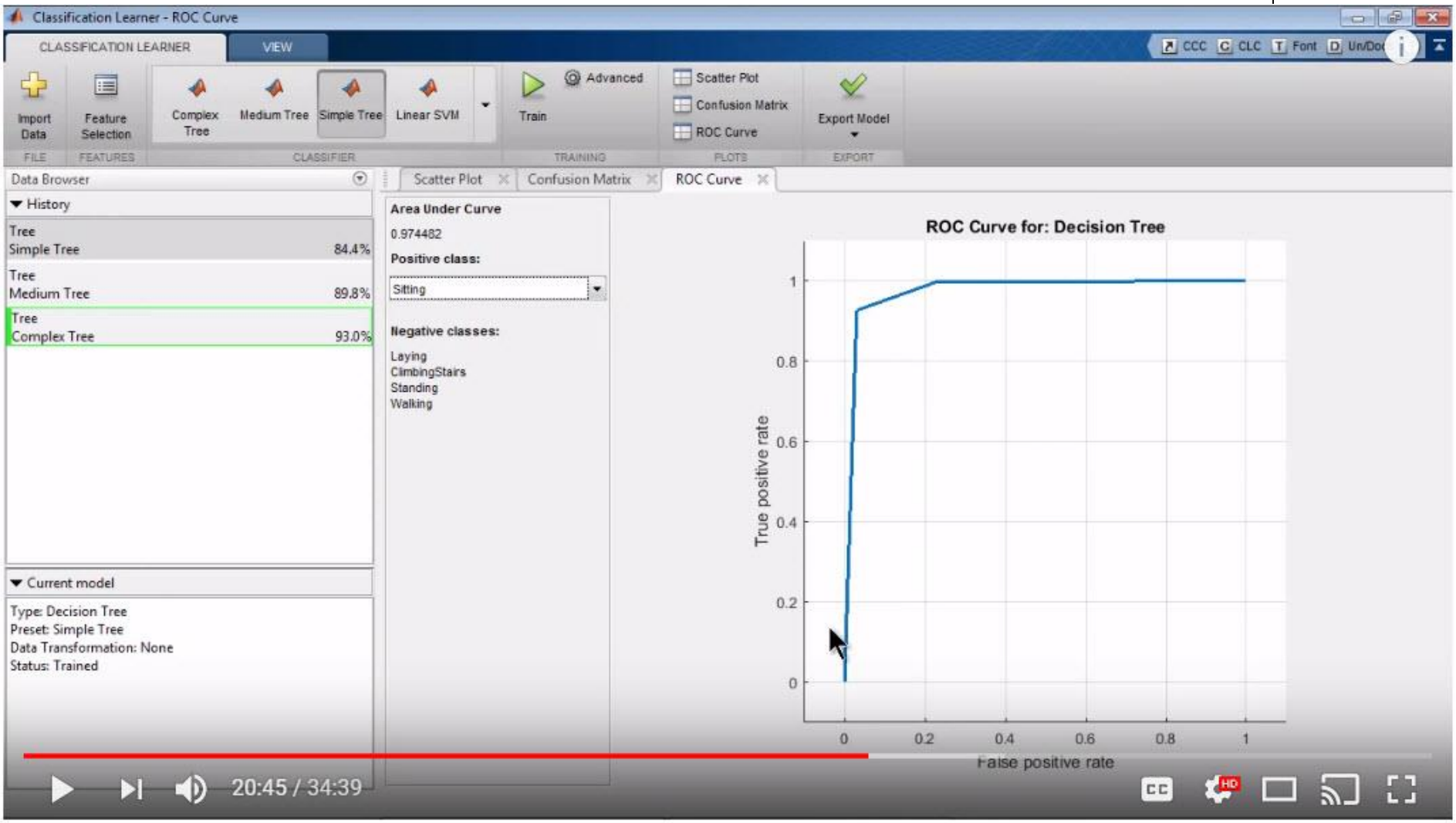

MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
W: ▸ Examples ▸ HumanActivity ▸
Current Folder
Name
html
Function
Wstd.m
Wpca1.m
Wmean.m
plotRawSensorData.m
plotModelResults.m
plotActivityResults.m
Wmean.m (Function)
Workspace
Name
Value
body_gyro_x_train 7352x128 doubl
body_gyro_y_train 7352x128 doubl
body_gyro_z_train 7352x128 doubl
rawSensorDataTrain 7352x6 table
total_acc_x_train 7352x128 doubl
total_acc_y_train 7352x128 doubl
Editor - W:\Examples\HumanActivity\Wmean.m
OpeningExample.m
Human_Activity_Learning.m
Wmean.m
1 function Y = Wmean(X)
2 % Copyright (c) 2015, MathWorks, Inc.
3 Y = mean(X,2);
4 end
Command History
Command Window
14:06 / 34:39

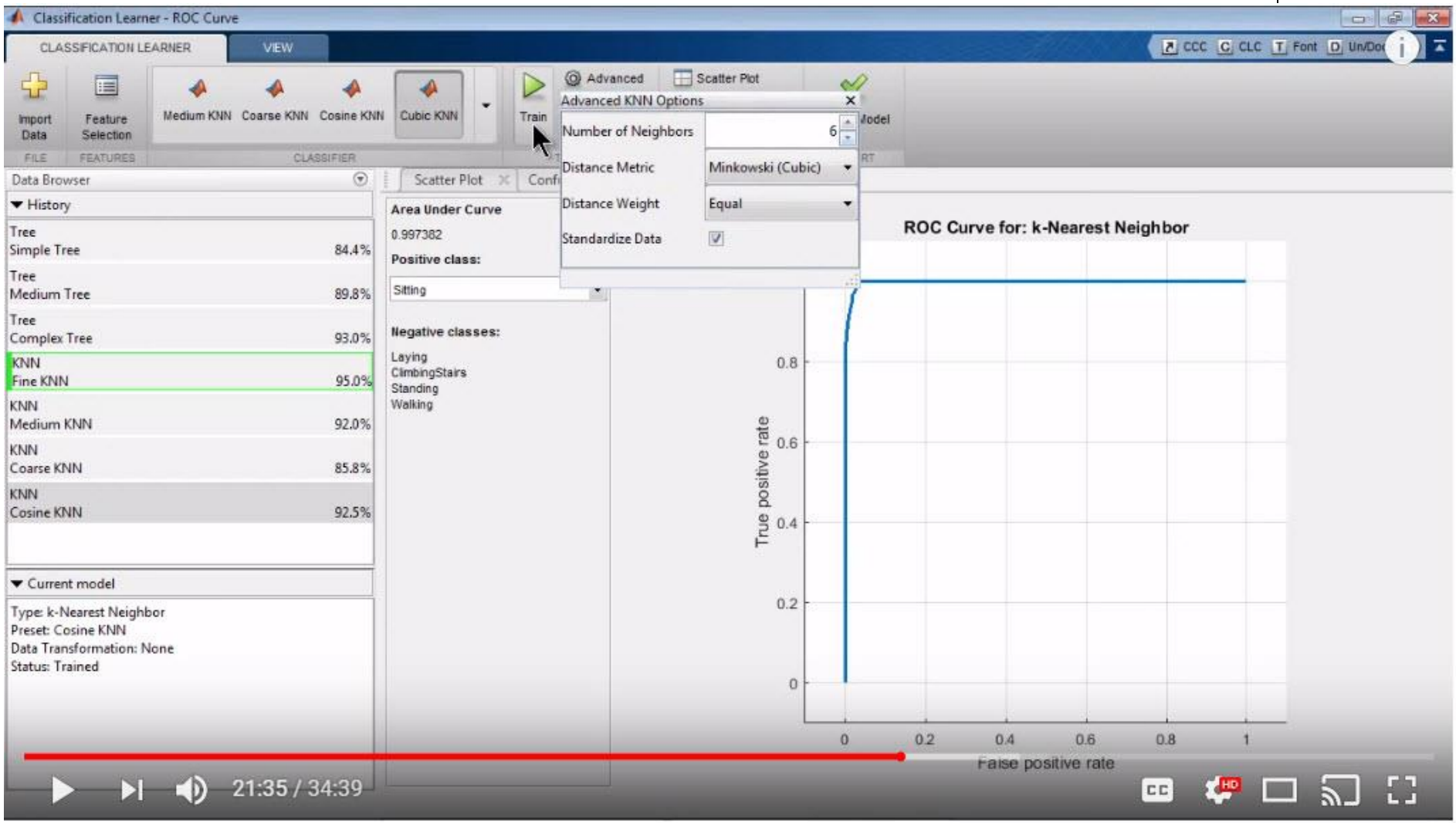

MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
Get More Apps
Install App
Package App
FILE
Search
FAVORITES
Curve Fitting
Classification Learner
Distribution Fitting
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Reco...
Neural Net Time Series
Signal Analysis
Filter Design & Analysis
MATLAB Coder
Report Generator
System Identification
MATH, STATISTICS AND OPTIMIZATION
Classification Learner
Curve Fitting
Distribution Fitting
MBC Model Fitting
MBC Optimization
MuPAD Notebook
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Reco...
Neural Net Time Series
Optimization
PDE
SIGNAL PROCESSING AND COMMUNICATIONS
TOP
Bit Error Rate Analysis
Eye Diagram Scope
Filter Builder
Filter Design & Analysis
LTE Downlink RMC Generator
LTE Test Model Generator
LTE Throughput ...
LTE Uplink RMC Generator
Radar Equation Calculator
Radar Waveform A...
RF Design & Analysis
Sensor Array Analyzer
Signal Analysis
Wavelet Design & An...
Window Design & An...
COMPUTATIONAL FINANCE
TOP
Financial Time Series
Current Folder
Name
html
Function
Wstd.m
Wpca1.m
Wmean.m
plotRawSens
plotModelR
plotActivityR
Wmean.m (Function)
Workspace
Name
body_gyro_x_tra
body_gyro_y_tra
body_gyro_z_tra
humanActivityDa
rawSensorDataT
T_mean
7352x6 table
T_pca
7352x6 table
33
% Step 1: Create a table
Command Window
Command History
15:32 / 34:39
script

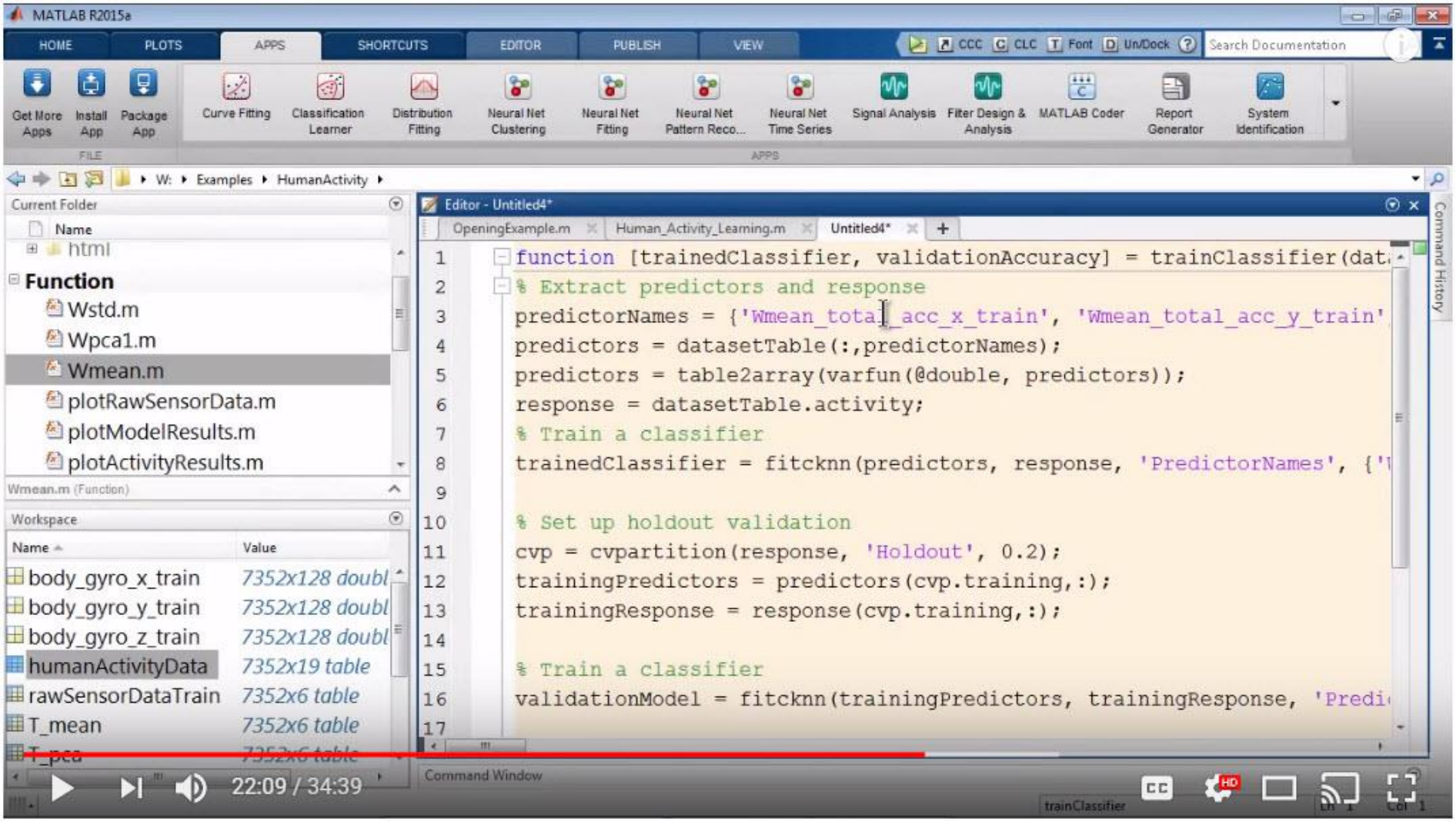

Set Up Classification
Step 1
Select dataset from MATLAB workspace.
T_mean
T_pca
T_stdv
humanActivityData
rawSensorDataTrain
body_gyro_x_train
body_gyro_y_train
body_gyro_z_train
total_acc_x_train
total_acc_y_train
total_acc_z_train
Use columns as variables
Use rows as variables
Step 2
Select predictors and response.
Name
Type
Range
Import as
Wmean_total_acc_x_t... double -0.3707 .. 1.05533 Predictor
Wmean_total_acc_y_t... double -0.494512 .. 1.005... Predictor
Wmean_total_acc_z_t... double -0.988372 .. 0.977... Predictor
Wmean_body_gyro_x... double -0.914161 .. 0.790... Predictor
Wmean_body_gyro_y... double -0.351097 .. 0.485... Predictor
Wmean_body_gyro_z... double -0.437807 .. 0.404... Predictor
Wstd_total_acc_x_train double 0.00134528 .. 0.6... Predictor
Wstd_total_acc_y_train double 0.00151609 .. 0.3... Predictor
Wstd_total_acc_z_train double 0.00298051 .. 0.3... Predictor
Wstd_body_gyro_x_tr... double 0.00169888 .. 1.7... Predictor
Wstd_body_gyro_y_tr... double 0.00201112 .. 1.5... Predictor
Wstd_body_gyro_z_tr... double 0.0021924 .. 0.97... Predictor
Wpca1_total_acc_x_t... double -13.2982 .. 2.85553 Predictor
Wpca1_total_acc_y_t... double -5.92044 .. 11.0487 Predictor
Wpca1_total_acc_z_t... double -12.1606 .. 10.0782 Predictor
Wpca1_body_gyro_x... double -9.18752 .. 8.79436 Predictor
Wpca1_body_gyro_y... double -10.0066 .. 10.9679 Predictor
Wpca1_body_gyro_z... double -9.08108 .. 9.22596 Predictor
activity categorical 5 unique Response
Step 3
Define validation method.
Cross Validation
Select a number of folds (or divisions) to partition the data into. Each fold is held out in turn for testing. The app trains a model for each fold using all the data outside the fold. The app tests each model performance using the data inside the fold, then calculates the average test error over all folds. This method gives a good estimate of the predictive accuracy of the final model trained with all the data. It requires multiple fits but makes efficient use of all
Holdout
Select a percentage of the data to use as a test set. The app trains the model on the training set and assesses the performance with the test set. Since the resulting model is based on only a portion of the data, it is recommended only for large data sets.
No Validation
Use all the data for training and compute the error rate on the same data. The estimated error rate is likely to be unrealistically low. The actual error rate when the model is used to predict new data is likely to be higher.
Percent Held Out: 25
Import Data
16:27 / 34:39
CC
HD


Classification Learner - Scatter Plot
CLASSIFICATION LEARNER
VIEW
Import Data
Feature Selection
Complex Tree
Medium Tree
Simple Tree
Linear SVM
Train
Advanced
Scatter Plot
Confusion Matrix
ROC Curve
Export Model
FILE
CLASSIFIER
TRAINING
PLOTS
EXPORT
Feature Selection
Wmean_total_acc_x_train
Wmean_total_acc_y_train
Wmean_total_acc_z_train
Wmean_body_gyro_x_train
Wmean_body_gyro_y_train
Wmean_body_gyro_z_train
Wstd_total_acc_x_train
Wstd_total_acc_y_train
Wstd_total_acc_z_train
Wstd_body_gyro_x_train
Wstd_body_gyro_y_train
Wstd_body_gyro_z_train
Wpca1_total_acc_x_train
Scatter Plot
Variable on X axis:
Wmean_total_acc_x_train
Variable on Y axis:
Wmean_total_acc_y_train
Legend
Observation from class
Laying
Sitting
ClimbingStairs
Show Classifier Results
Classifier Results
< Model not trained >
Current model
Type:
Preset:
Data Transformation: None
Status: Untrained
Scatter Plot of humanActivityData
Wmean_total_acc_y_train
Wmean_total_acc_x_train
17:38 / 34:39


Classification Learner - Confusion Matrix
CLASSIFICATION LEARNER
VIEW
Import Data
Feature Selection
Complex Tree
Medium Tree
Simple Tree
Linear SVM
Train
Advanced
Scatter Plot
Confusion Matrix
ROC Curve
Export Model
FILE
FEATURES
CLASSIFIER
TRAINING
PLOTS
EXPORT
Data Browser
History
Tree
Simple Tree 84.4%
Tree
Medium Tree 89.8%
Tree
Complex Tree 93.0%
Current model
Type: Decision Tree
Preset: Complex Tree
Data Transformation: None
Status: Trained
Scatter Plot
Confusion Matrix
Overall Accuracy
93.0%
Overall Error
7.0%
Summarize
Per true class
View percentages per true class including True Positive Rates (TPR) and False Negative Rates (FNR).
Per predicted class
View percentages per predicted class including Positive Predictive Values (PPV) and False Discovery Rates (FDR).
Overall
View percentages over the entire confusion matrix.
Confusion Matrix for: Decision Tree
True class
Laying
Sitting
ClimbingStairs
Standing
Walking
282 100%
0 0.0%
0 0.0%
0 0.0%
0 0.0%
0 0.0%
248 96.5%
0 0.0%
8 3.1%
1 0.4%
0 0.0%
0 0.0%
366 89.1%
0 0.0%
45 10.9%
0 0.0%
20 7.3%
0 0.0%
254 92.7%
0 0.0%
0 0.0%
0 0.0%
29 11.8%
0 0.0%
217 88.2%
100% 0.0%
96.5% 3.5%
89.1% 10.9%
92.7% 7.3%
88.2% 11.8%
TPR / FNR
Predicted class
19:45 / 34:39


Classification Learner - ROC Curve
CLASSIFICATION LEARNER
VIEW
Import Data
Feature Selection
Complex Tree
Medium Tree
Simple Tree
Linear SVM
Train
Advanced
Scatter Plot
Confusion Matrix
ROC Curve
Export Model
FILE
FEATURES
CLASSIFIER
TRAINING
PLOTS
EXPORT
Data Browser
History
Tree
Simple Tree 84.4%
Tree
Medium Tree 89.8%
Tree
Complex Tree 93.0%
Current model
Type: Decision Tree
Preset: Simple Tree
Data Transformation: None
Status: Trained
Scatter Plot
Confusion Matrix
ROC Curve
Area Under Curve
0.974482
Positive class:
Sitting
Negative classes:
Laying
ClimbingStairs
Standing
Walking
ROC Curve for: Decision Tree
True positive rate
False positive rate
0
0.2
0.4
0.6
0.8
1
20:45 / 34:39


Classification Learner - ROC Curve
CLASSIFICATION LEARNER
VIEW
Import Data
Feature Selection
Medium KNN
Coarse KNN
Cosine KNN
Cubic KNN
Train
Advanced
Scatter Plot
Advanced KNN Options
Number of Neighbors
6
Distance Metric
Minkowski (Cubic)
Distance Weight
Equal
Standardize Data
Data Browser
History
Tree
Simple Tree 84.4%
Tree
Medium Tree 89.8%
Tree
Complex Tree 93.0%
KNN
Fine KNN 95.0%
KNN
Medium KNN 92.0%
KNN
Coarse KNN 85.8%
KNN
Cosine KNN 92.5%
Current model
Type: k-Nearest Neighbor
Preset: Cosine KNN
Data Transformation: None
Status: Trained
Area Under Curve
0.997382
Positive class:
Sitting
Negative classes:
Laying
ClimbingStairs
Standing
Walking
ROC Curve for: k-Nearest Neighbor
True positive rate
False positive rate
0
0.2
0.4
0.6
0.8
1
21:35 / 34:39


MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
Get More Apps
Install App
Package App
Curve Fitting
Classification Learner
Distribution Fitting
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Reco...
Neural Net Time Series
Signal Analysis
Filter Design & Analysis
MATLAB Coder
Report Generator
System Identification
W: ▸ Examples ▸ HumanActivity ▸
Current Folder
Name
Function
Wstd.m
Wpca1.m
Wmean.m
plotRawSensorData.m
plotModelResults.m
plotActivityResults.m
Wmean.m (Function)
Workspace
Name
Value
body_gyro_x_train 7352x128 doubl
body_gyro_y_train 7352x128 doubl
body_gyro_z_train 7352x128 doubl
humanActivityData 7352x19 table
rawSensorDataTrain 7352x6 table
T_mean 7352x6 table
Editor - Untitled4*
OpeningExample.m
Human_Activity_Learning.m
Untitled4*
1 function [trainedClassifier, validationAccuracy] = trainClassifier(dat
2 % Extract predictors and response
3 predictorNames = {'Wmean_total_acc_x_train', 'Wmean_total_acc_y_train'
4 predictors = datasetTable(:,predictorNames);
5 predictors = table2array(varfun(@double, predictors));
6 response = datasetTable.activity;
7 % Train a classifier
8 trainedClassifier = fitcknn(predictors, response, 'PredictorNames', {'
9
10 % Set up holdout validation
11 cvp = cvpartition(response, 'Holdout', 0.2);
12 trainingPredictors = predictors(cvp.training,:);
13 trainingResponse = response(cvp.training,:);
14
15 % Train a classifier
16 validationModel = fitcknn(trainingPredictors, trainingResponse, 'Predi
17
Command History
Command Window
22:09 / 34:39
trainClassifier

MathWorks

Example 2: Real-time Car Identification Using Images

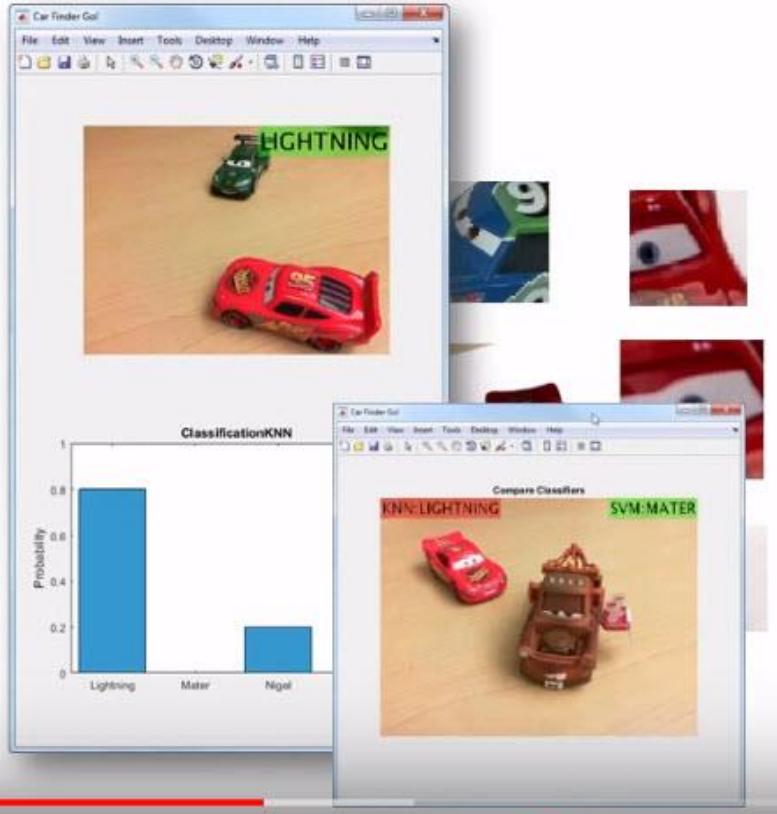
Objective: Train a classifier to identify car type from a webcam video

Data:

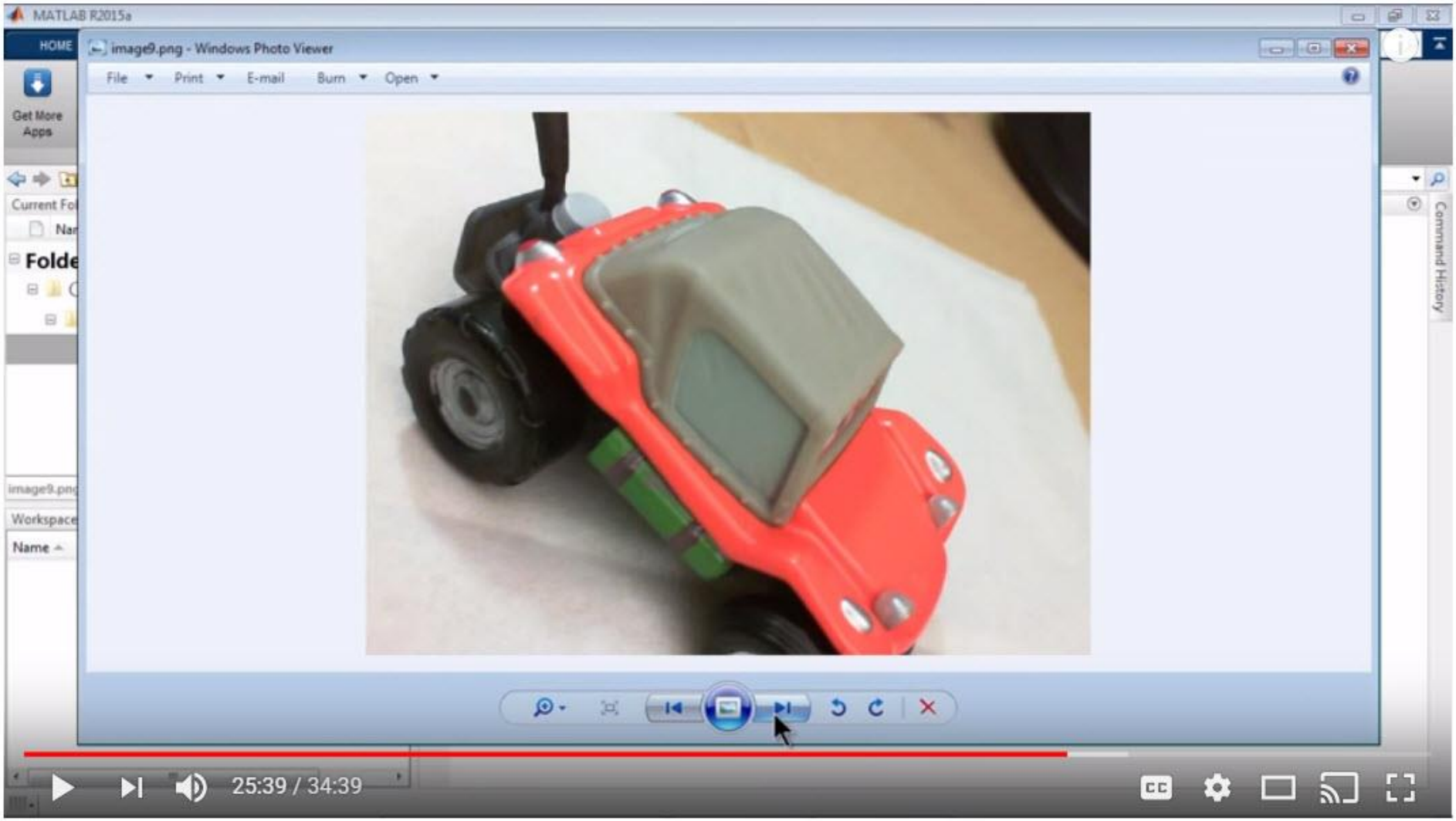
Predictors	Several images of cars: 
Response	**NIGEL, LIGHTNING, SANDDUNE, MATER**

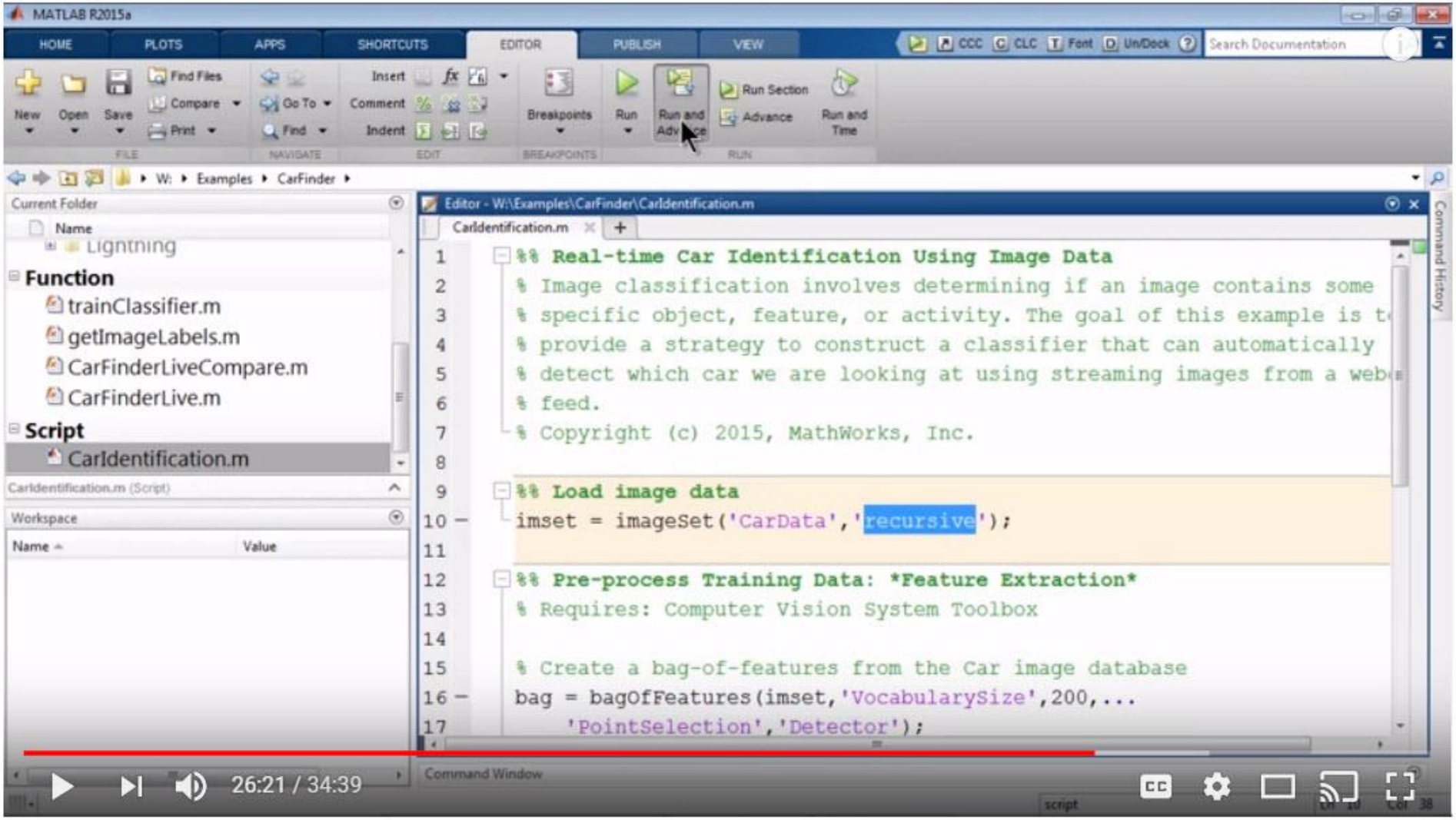
Approach:

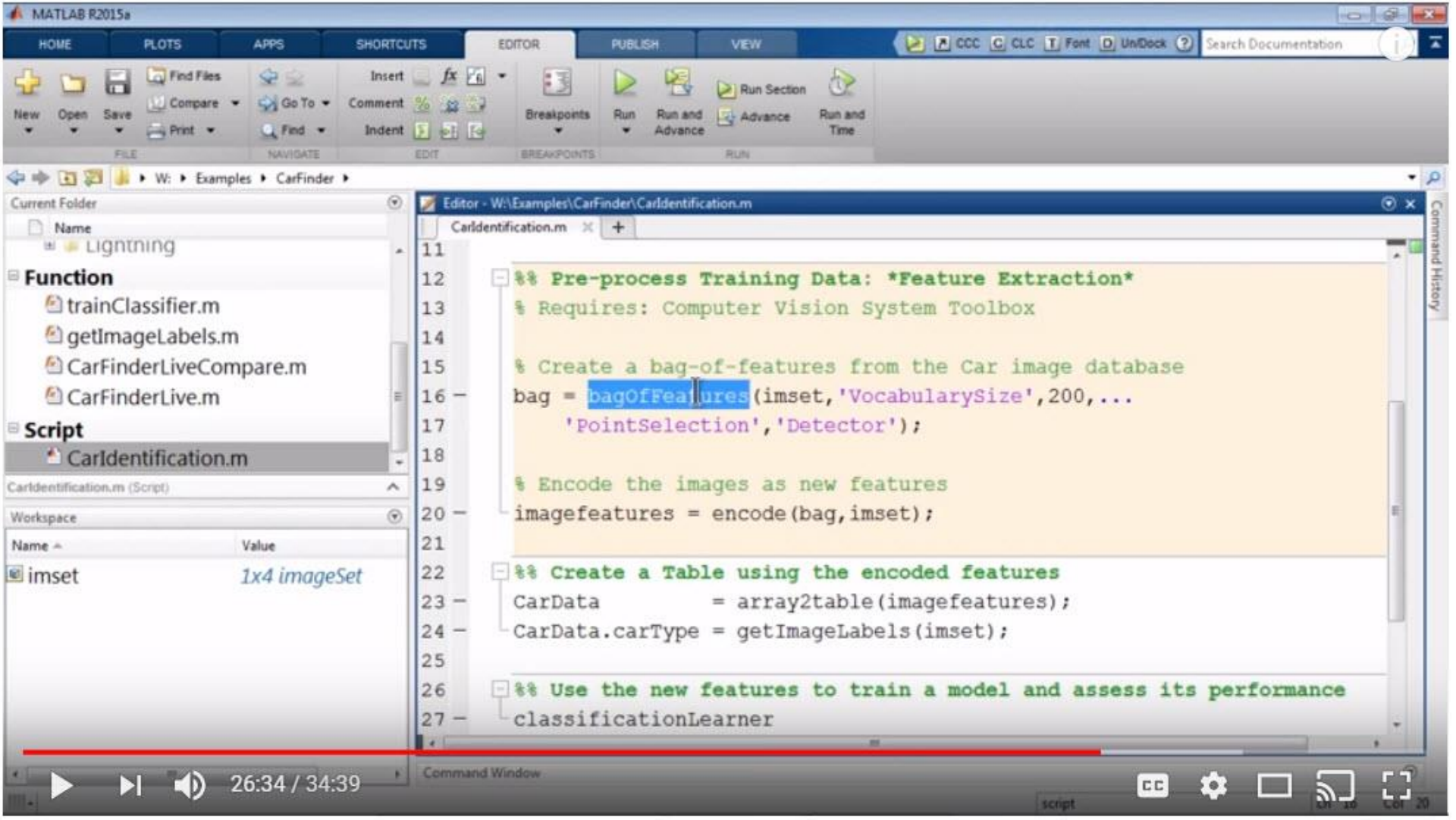
- Extract features using Bag-of-words
- Train and compare classifiers
- Classify streaming video from a webcam



24:27 / 34:39


MATLAB R2015a
HOME
Get More Apps
image9.png - Windows Photo Viewer
File
Print
E-mail
Burn
Open
Current Fo
Folde
Workspace
Name
Command History
25:39 / 34:39

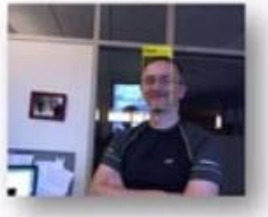

MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
W: ▸ Examples ▸ CarFinder ▸
Current Folder
Name
Function
trainClassifier.m
getImageLabels.m
CarFinderLiveCompare.m
CarFinderLive.m
Script
CarIdentification.m
CarIdentification.m (Script)
Workspace
Name
Value
Editor - W:\Examples\CarFinder\CarIdentification.m
CarIdentification.m
1 %% Real-time Car Identification Using Image Data
2 % Image classification involves determining if an image contains some
3 % specific object, feature, or activity. The goal of this example is t
4 % provide a strategy to construct a classifier that can automatically
5 % detect which car we are looking at using streaming images from a web
6 % feed.
7 % Copyright (c) 2015, MathWorks, Inc.
8
9 %% Load image data
10 imset = imageSet('CarData','recursive');
11
12 %% Pre-process Training Data: *Feature Extraction*
13 % Requires: Computer Vision System Toolbox
14
15 % Create a bag-of-features from the Car image database
16 bag = bagOfFeatures(imset,'VocabularySize',200,...
17 'PointSelection','Detector');
Command Window
Command History
26:21 / 34:39
CC


MATLAB R2015a
HOME
PLOTS
APPS
SHORTCUTS
EDITOR
PUBLISH
VIEW
Search Documentation
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
W: ▸ Examples ▸ CarFinder ▸
Current Folder
Name
Function
trainClassifier.m
getImageLabels.m
CarFinderLiveCompare.m
CarFinderLive.m
Script
CarIdentification.m
Workspace
Name
Value
imset
1x4 imageSet
Editor - W:\Examples\CarFinder\CarIdentification.m
CarIdentification.m
%% Pre-process Training Data: *Feature Extraction*
% Requires: Computer Vision System Toolbox
% Create a bag-of-features from the Car image database
bag = bagOfFeatures(imset,'VocabularySize',200,...
'PointSelection','Detector');
% Encode the images as new features
imagefeatures = encode(bag,imset);
%% Create a Table using the encoded features
CarData = array2table(imagefeatures);
CarData.carType = getImageLabels(imset);
%% Use the new features to train a model and assess its performance
classificationLearner
Command Window
Command History
26:34 / 34:39

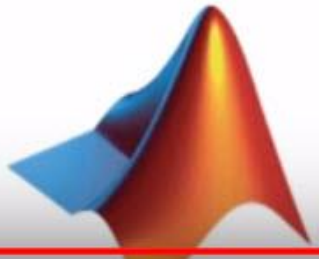
MathWorks

Key Takeaways

- Consider Machine Learning when:
  - Hand written rules and equations are too complex
    - *Face recognition, speech recognition, recognizing patterns*
  - Rules of a task are constantly changing
    - *Fraud detection from transactions, anomaly in sensor data*
  - Nature of the data changes and the program needs to adapt
    - *Automated trading, energy demand forecasting, predicting shopping trends*



- MATLAB for Machine Learning



Email me if you have further questions