Chapter 04a - Machine Learning with MATLAB.

Author: Yijie ZHENG (Amy)

Reference files for this chapter:

- ../Test_Data/rawSensorData_train.mat
- ../Test_Data/rawSensorData_test.mat
- Wstd.m
- Wmean.m
- plotRawSensorData.m

Machine Learning with Matlab

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

https://uk.mathworks.com/help/stats/machine-learning-in-matlab.html

Train Classification Models in Classification Learner App

Description of the Data (rawSensorData_train/test.mat)

`The dataset consists of accelerometer and gyroscope data captured at 50Hz. The activities performed by the subject include: 'Walking', 'ClimbingStairs', 'Sitting', 'Standing',and 'Laying'`

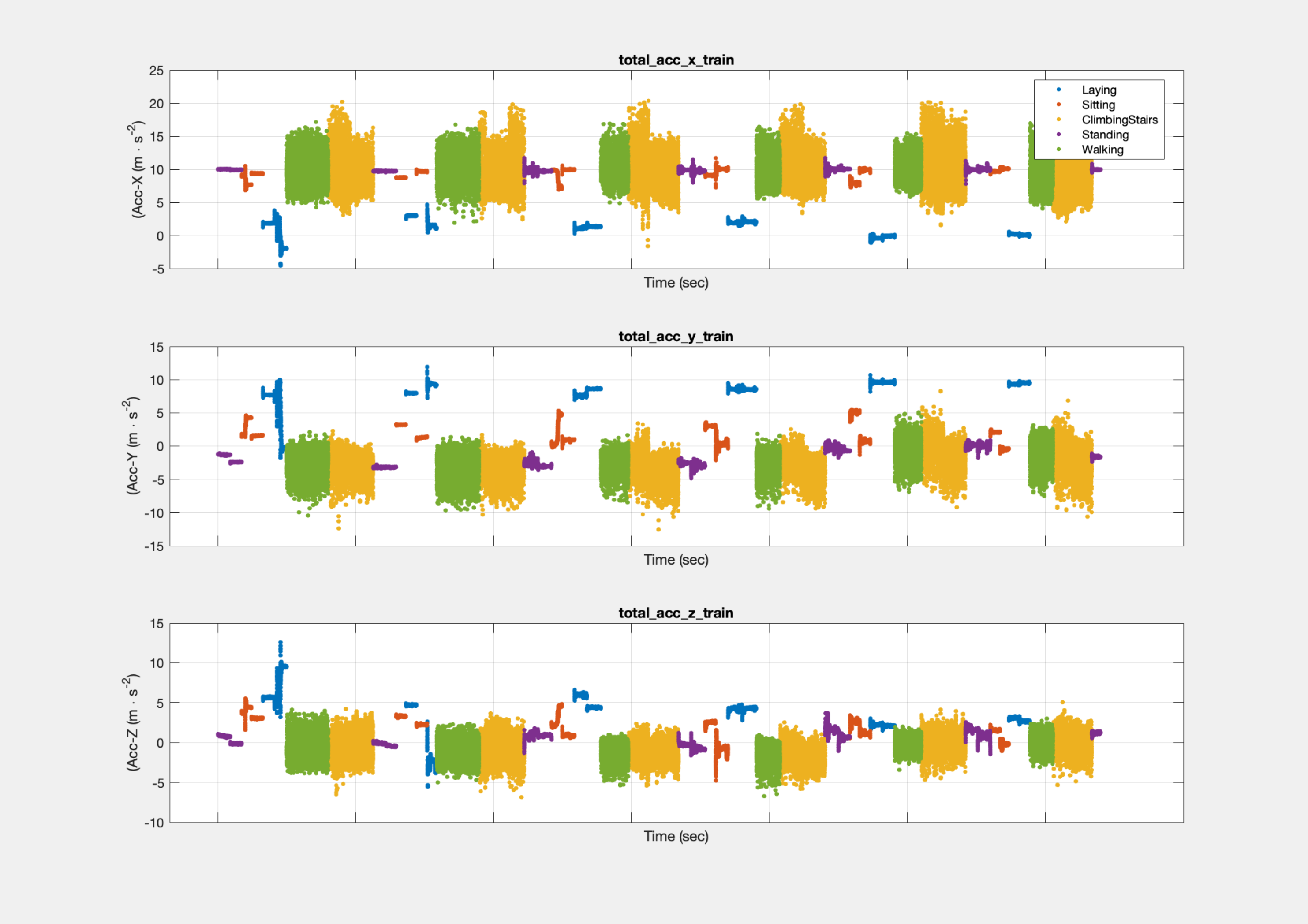
1. *total_acc_(x/y/z)_train :* Raw accelerometer sensor data
2. *body_gyro_(x/y/z)_train :* Raw gyroscope sensor data
3. *trainActivity :* Training data labels
4. *testActivity :* Test data labels

Load Training Data

```
load rawSensorData_train.mat
```

Display data summary

```
plotRawSensorData(total_acc_x_train, total_acc_y_train, ...
    total_acc_z_train,trainActivity,1000)
```



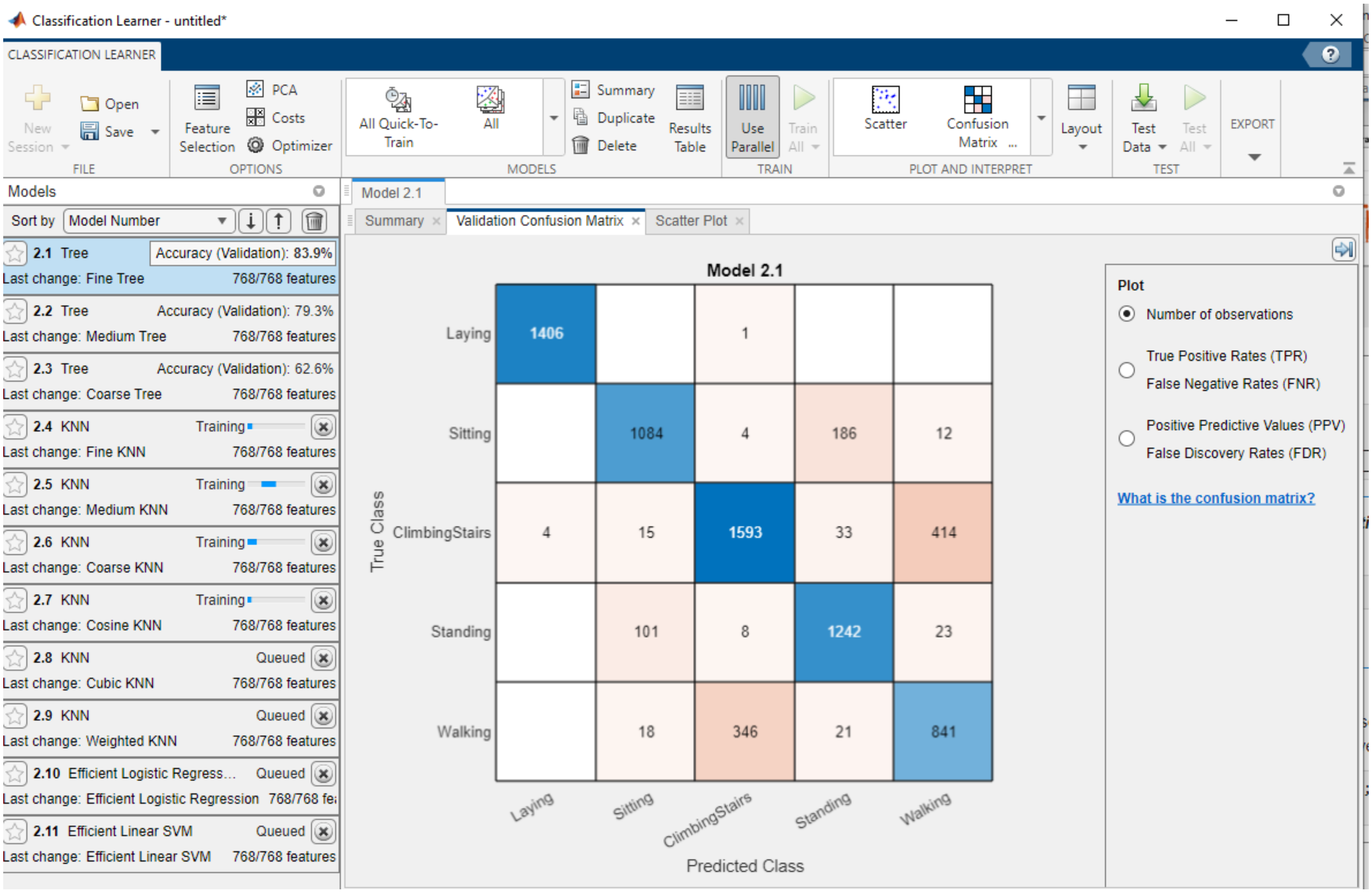
Create Table variable

```
rawSensorDataTrain = table(...
    total_acc_x_train, total_acc_y_train, total_acc_z_train, ...
    body_gyro_x_train, body_gyro_y_train, body_gyro_z_train);
```

Train a model and assess its performance using Classification Learner

Train on raw data

```
classificationLearner
```



2.1 Tree Accuracy (Validation): 83.9%
Last change: Fine Tree 768/768 features

2.2 Tree Accuracy (Validation): 79.3%
Last change: Medium Tree 768/768 features

2.3 Tree Accuracy (Validation): 62.6%
Last change: Coarse Tree 768/768 features

2.4 KNN Training
Last change: Fine KNN 768/768 features

2.5 KNN Training
Last change: Medium KNN 768/768 features

2.6 KNN Training
Last change: Coarse KNN 768/768 features

2.7 KNN Training
Last change: Cosine KNN 768/768 features

2.8 KNN Queued
Last change: Cubic KNN 768/768 features

2.9 KNN Queued
Last change: Weighted KNN 768/768 features

2.10 Efficient Logistic Regress... Queued
Last change: Efficient Logistic Regression 768/768 fe

2.11 Efficient Linear SVM Queued
Last change: Efficient Linear SVM 768/768 features

Model 2.1

True Class \ Predicted Class	Laying	Sitting	ClimbingStairs	Standing	Walking
Laying	1406		1		
Sitting		1084	4	186	12
ClimbingStairs	4	15	1593	33	414
Standing		101	8	1242	23
Walking		18	346	21	841

Plot

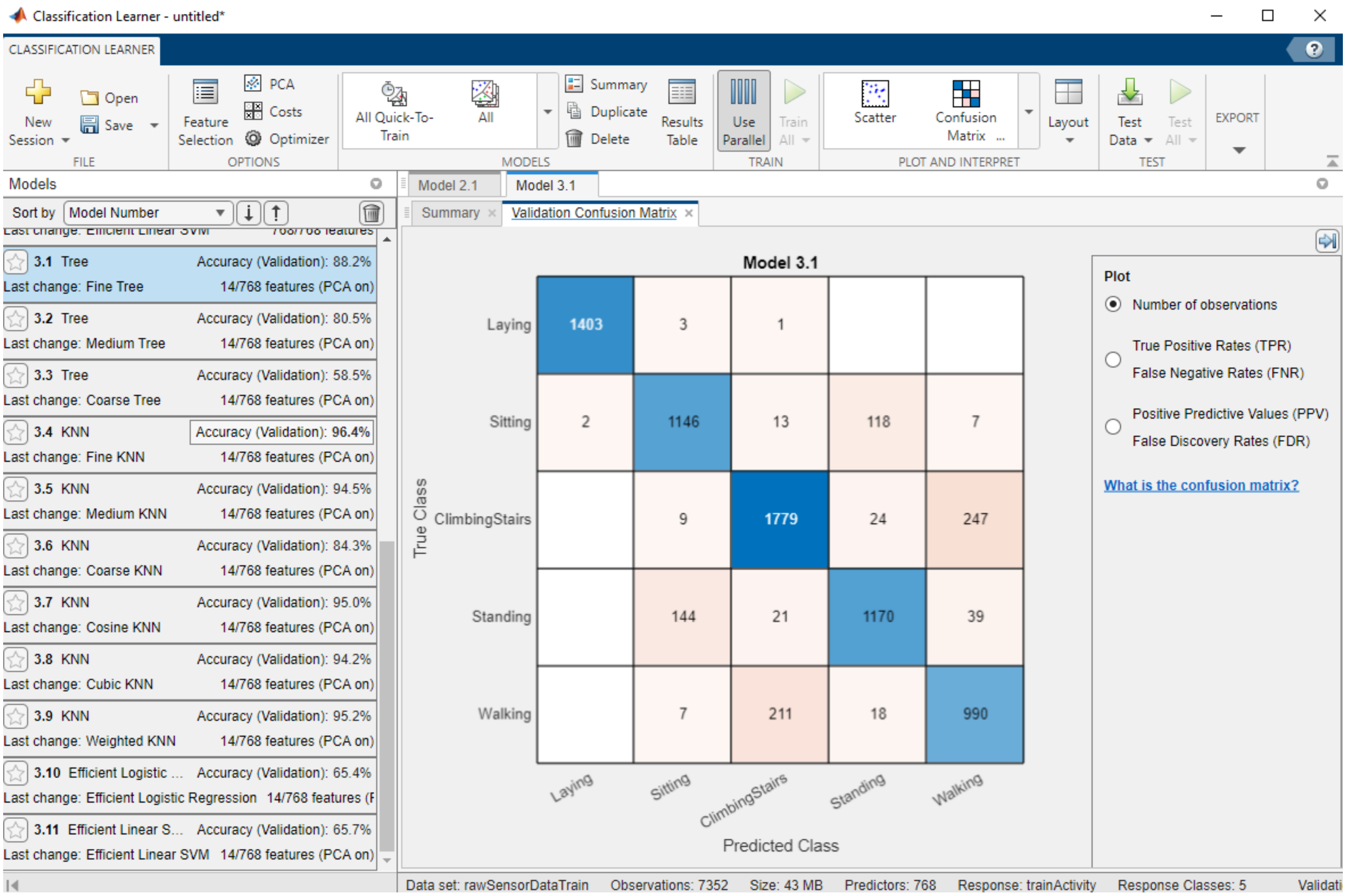
● Number of observations

○ True Positive Rates (TPR)
False Negative Rates (FNR)

○ Positive Predictive Values (PPV)
False Discovery Rates (FDR)

What is the confusion matrix?

Enable PCA



Last change: Efficient Linear SVM 768/768 features

3.1 Tree Accuracy (Validation): 88.2%
Last change: Fine Tree 14/768 features (PCA on)

3.2 Tree Accuracy (Validation): 80.5%
Last change: Medium Tree 14/768 features (PCA on)

3.3 Tree Accuracy (Validation): 58.5%
Last change: Coarse Tree 14/768 features (PCA on)

3.4 KNN Accuracy (Validation): 96.4%
Last change: Fine KNN 14/768 features (PCA on)

3.5 KNN Accuracy (Validation): 94.5%
Last change: Medium KNN 14/768 features (PCA on)

3.6 KNN Accuracy (Validation): 84.3%
Last change: Coarse KNN 14/768 features (PCA on)

3.7 KNN Accuracy (Validation): 95.0%
Last change: Cosine KNN 14/768 features (PCA on)

3.8 KNN Accuracy (Validation): 94.2%
Last change: Cubic KNN 14/768 features (PCA on)

3.9 KNN Accuracy (Validation): 95.2%
Last change: Weighted KNN 14/768 features (PCA on)

3.10 Efficient Logistic ... Accuracy (Validation): 65.4%
Last change: Efficient Logistic Regression 14/768 features (F

3.11 Efficient Linear S... Accuracy (Validation): 65.7%
Last change: Efficient Linear SVM 14/768 features (PCA on)

Model 3.1

True Class \ Predicted Class	Laying	Sitting	ClimbingStairs	Standing	Walking
Laying	1403	3	1		
Sitting	2	1146	13	118	7
ClimbingStairs		9	1779	24	247
Standing		144	21	1170	39
Walking		7	211	18	990

Plot
- ● Number of observations
- ○ True Positive Rates (TPR)
False Negative Rates (FNR)
- ○ Positive Predictive Values (PPV)
False Discovery Rates (FDR)

Data set: rawSensorDataTrain Observations: 7352 Size: 43 MB Predictors: 768 Response: trainActivity Response Classes: 5 Validati

Pre-process Training Data: Feature Extraction

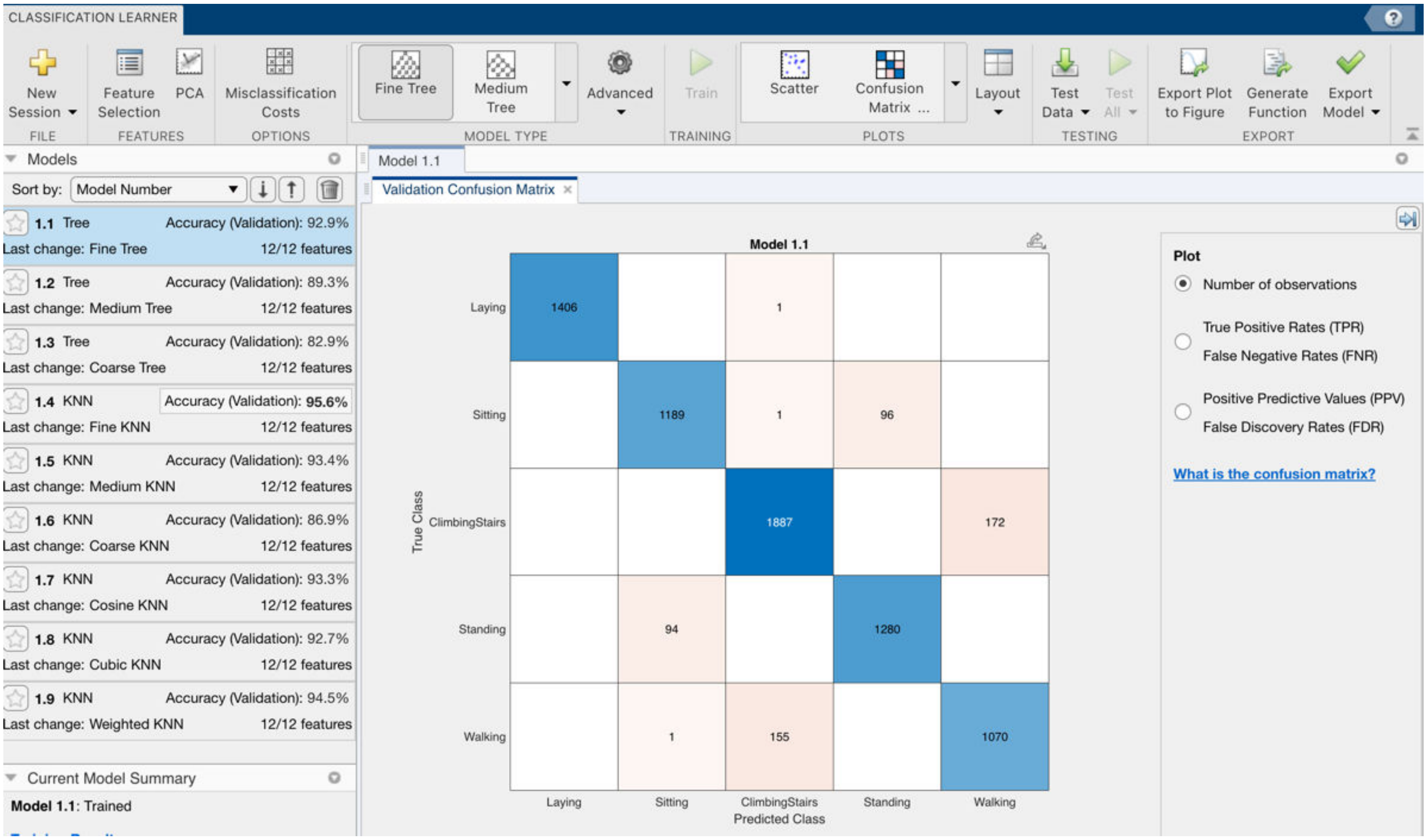
Lets start with a simple preprocessing technique. Since the raw sensor data contain fixed-width sliding windows of 2.56sec (128 readings/window) lets start with a simple average feature for every 128 points

```
%% calculate the mean value
T_mean = varfun(@Wmean,rawSensorDataTrain);

%% calculate the std value
T_stdv = varfun(@Wstd,rawSensorDataTrain);


%% update the table
humanActivityData = [T_mean, T_stdv];
humanActivityData.activity = trainActivity;
humanActivityData.Properties.VariableNames = {'input1', 'input2', 'input3',...
    'input4','input5','input6','input7','input8','input9','input10',...
    'input11','input12','output'};
```

Train a model and assess its performance using Classification Learner

Train on pre-processed data

```
classificationLearner
```



Sort by: Model Number

1.1 Tree Accuracy (Validation): 92.9%
Last change: Fine Tree 12/12 features

1.2 Tree Accuracy (Validation): 89.3%
Last change: Medium Tree 12/12 features

1.3 Tree Accuracy (Validation): 82.9%
Last change: Coarse Tree 12/12 features

1.4 KNN Accuracy (Validation): **95.6%**
Last change: Fine KNN 12/12 features

1.5 KNN Accuracy (Validation): 93.4%
Last change: Medium KNN 12/12 features

1.6 KNN Accuracy (Validation): 86.9%
Last change: Coarse KNN 12/12 features

1.7 KNN Accuracy (Validation): 93.3%
Last change: Cosine KNN 12/12 features

1.8 KNN Accuracy (Validation): 92.7%
Last change: Cubic KNN 12/12 features

1.9 KNN Accuracy (Validation): 94.5%
Last change: Weighted KNN 12/12 features

Current Model Summary

Model 1.1: Trained

Model 1.1

True Class \ Predicted Class	Laying	Sitting	ClimbingStairs	Standing	Walking
Laying	1406		1		
Sitting		1189	1	96	
ClimbingStairs			1887		172
Standing		94		1280	
Walking		1	155		1070

Plot

● Number of observations

○ True Positive Rates (TPR)
False Negative Rates (FNR)

○ Positive Predictive Values (PPV)
False Discovery Rates (FDR)

What is the confusion matrix?

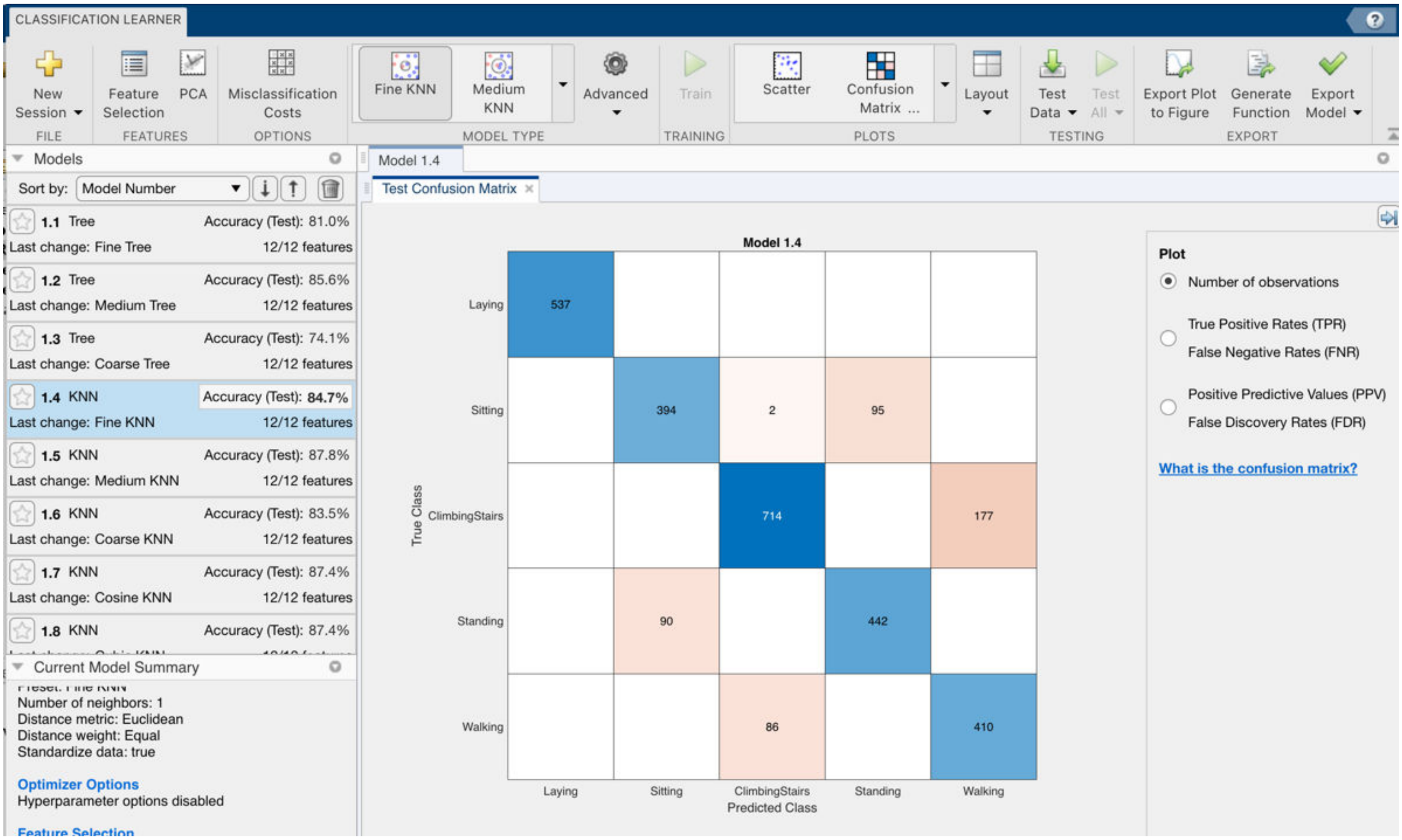
Export trained model

Name: 'trainedModel'

Make Predictions for New Data Using Exported Model

```
%% load test data
load rawSensorData_test
% Step 1: Create the table
rawSensorDataTest = table(...
    total_acc_x_test, total_acc_y_test, total_acc_z_test, ...
    body_gyro_x_test, body_gyro_y_test, body_gyro_z_test);

% Step 2: Extract features from raw sensor data
T_mean = varfun(@Wmean, rawSensorDataTest);
T_stdv = varfun(@Wstd,rawSensorDataTest);

humanActivityData2 = [T_mean, T_stdv];
humanActivityData2.activity = testActivity;
humanActivityData2.Properties.VariableNames = {'input1', 'input2', 'input3',...
    'input4','input5','input6','input7','input8','input9','input10',...
    'input11','input12','output'};
```

Use Classification Learner App



Models

Sort by: Model Number

1.1 Tree Accuracy (Test): 81.0%
Last change: Fine Tree 12/12 features

1.2 Tree Accuracy (Test): 85.6%
Last change: Medium Tree 12/12 features

1.3 Tree Accuracy (Test): 74.1%
Last change: Coarse Tree 12/12 features

1.4 KNN Accuracy (Test): **84.7%**
Last change: Fine KNN 12/12 features

1.5 KNN Accuracy (Test): 87.8%
Last change: Medium KNN 12/12 features

1.6 KNN Accuracy (Test): 83.5%
Last change: Coarse KNN 12/12 features

1.7 KNN Accuracy (Test): 87.4%
Last change: Cosine KNN 12/12 features

1.8 KNN Accuracy (Test): 87.4%
Last change: Cubic KNN 12/12 features

Current Model Summary

Preset: Fine KNN
Number of neighbors: 1
Distance metric: Euclidean
Distance weight: Equal
Standardize data: true

Optimizer Options
Hyperparameter options disabled

Feature Selection

Model 1.4

True Class \ Predicted Class	Laying	Sitting	ClimbingStairs	Standing	Walking
Laying	537				
Sitting		394	2	95	
ClimbingStairs			714		177
Standing		90		442	
Walking			86		410

Plot

● Number of observations

○ True Positive Rates (TPR)
False Negative Rates (FNR)

○ Positive Predictive Values (PPV)
False Discovery Rates (FDR)

What is the confusion matrix?

Use function 'predictFcn'

```
[yfit,scores] = trainedModel.predictFcn(humanActivityData2)
trueLabels = humanActivityData2.output;
figure;
confusionchart(trueLabels, yfit);
title('Confusion Matrix');
```