```
%{
Support vector machine
%}

clc

% ######### Set aside some of the data for testing ##########

% We want to shuffle both inputs and outputs while preserving the
% correlation
p = randperm(length(final_inputs_svm));
random_final_inputs = final_inputs_svm(p, :);
random_final_targets = final_targets_svm(p, :);

% set some percentage of it aside for testing
test_percent = 85;
test_element_count =
 uint32((test_percent/100)*length(random_final_inputs));

% Define which features to include in the input set.
train_inputs = random_final_inputs(1:end-test_element_count,:);    %
 Take all the rows, and all the 10 features as inputs. Could also use:
 inputs = dataSet(:,1:end-2).
test_inputs= random_final_inputs(end-test_element_count+1:end,:);

% Define the target set
train_targets = random_final_targets(1:end-test_element_count, :);
test_targets = random_final_targets(end-test_element_count+1:end,:);


% ######### Model Training ##################################

%Create a SVM Model template to fit into fitcecoc()
t = templateSVM('Standardize',true);

SVMModel_Multi = fitcecoc(train_inputs,train_targets,'Learners',t);

SVMModel_Multi.ClassNames

CodingMat = SVMModel_Multi.CodingMatrix;



% ######### Model Evaluation ##################################

% predict using the test data extracted earlier
targets_from_test_prediction =
 predict(SVMModel_Multi,test_inputs,'Verbose',1);

% print out the Binary Loss function has been applied to the model
 ('hinge' by default)
SVMModel_Multi.BinaryLoss
```

```
% Plot Confusion Matrix
confusionchart(test_targets,targets_from_test_prediction);


%{
now I have created new method for labelling data for SVMs as it is a
 different process
than that for ANNs. Created a multiclass SVM classifier for the data
 and
currently receiving 100% classification accuracy using the hinge
 binary loss
function. Will switch to assessing the classifier performance in a
 more robust way
by obtaining the Posterior Probabilities.
Explained
openExample('stats/
EstimatePosteriorProbabilitiesUsingECOCClassifiersExample')
https://www.youtube.com/watch?v=FpKiHpYnY_I
%}
```

Published with MATLAB® R2020a