

Classifying Urban sounds using Machine Learning and Deep Learning

3 Model Training and Evaluation

Before training the model, I will convert the label in dataTrain_Validation and dataTest from categorical array to string array. It is because categorical array does not accepted by code generation using MATLAB coder in deployment

```
dataTrain_Validation(:,15)=string(dataTrain_Validation(:,14));  
dataTrain_Validation(:,14)=[];  
dataTrain_Validation.Properties.VariableNames{14} = 'Label';  
  
dataTest(:,15)=string(dataTest(:,14));  
dataTest(:,14)=[];  
dataTest.Properties.VariableNames{14} = 'Label';
```

Train the model Using Classification Learner APP

Interactively train, validate, and tune classification models









Choose among various algorithms to train and validate classification models for binary or multiclass problems. After training multiple models, compare their validation errors side-by-side, and then choose the best model. To help you decide which algorithm to use, see [Train Classification Models in Classification Learner App](#).

```
classificationLearner
```

Taking into Consideration of Choosing the Right Model for you.

<https://www.mathworks.com/help/stats/choose-a-classifier.html>

Characteristics of Classifier Types

Classifier	Prediction Speed	Memory Usage
Decision Trees 	Fast	Small
Discriminant Analysis 	Fast	Small for linear quadratic
Logistic Regression 	Fast	Medium
Support Vector Machines  <div>  Note Supports C code generation for prediction. </div>	Medium for linear Slow for others	Medium for linear All others: medium multiclass, large
Nearest Neighbor Classifiers 	Slow for cubic Medium for others	Medium
Ensemble Classifiers 	Fast to medium depending on choice of algorithm	Low to high depending on choice of algorithm
Naive Bayes Classifiers 	Medium for simple distributions Slow for kernel distributions or high- dimensional data	Small for simple Medium for kernel or high-dimensional

Training Model in Classification Learner App

- 1) Import Data with 10 folds

New Session

Data set

Workspace Variable

dataTrain_Validation 2620x14 table

Response

Label string 10 unique

Predictors

	Name	Type	Range
<input checked="" type="checkbox"/>	Signal_stats_Col1_ShapeFac...	double	1.05916 .. 7.35721
<input checked="" type="checkbox"/>	Signal_stats_Col1_Kurtosis	double	1.26461 .. 1705.93
<input checked="" type="checkbox"/>	Signal_stats_Col1_Clearance...	double	1.57408 .. 1233.49
<input checked="" type="checkbox"/>	Signal_stats_Col1_ImpulseFa...	double	1.49142 .. 352.781
<input checked="" type="checkbox"/>	Signal_stats_Col1_CrestFactor	double	1.38814 .. 71.8354
<input checked="" type="checkbox"/>	Signal_stats_Col1_PeakValue	double	0.00109863 .. 1
<input checked="" type="checkbox"/>	Signal_stats_Col1_SNR	double	-28.6255 .. 25.0002
<input checked="" type="checkbox"/>	Signal_stats_Col1_SINAD	double	-28.6255 .. 24.9963
<input checked="" type="checkbox"/>	Signal_ps_spec_Col1_Band...	double	1.04482e-07 .. 1.04199
<input checked="" type="checkbox"/>	Signal_stats_Col1_Std	double	0.000183097 .. 0.709569
<input checked="" type="checkbox"/>	Signal_stats_Col1_RMS	double	0.000183735 .. 0.720389
<input checked="" type="checkbox"/>	Signal_ps_spec_Col1_Peak...	double	0.0035834 .. 2.66819
<input checked="" type="checkbox"/>	Signal_ps_spec_Col1_Peak...	double	1.48108e-10 .. 19.3277

☐ Label string 40 unique

[How to prepare data](#)

Validation

☒ **Cross-Validation**

Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.

Cross-validation folds: 10 folds

☐ **Holdout Validation**

Recommended for large data sets.

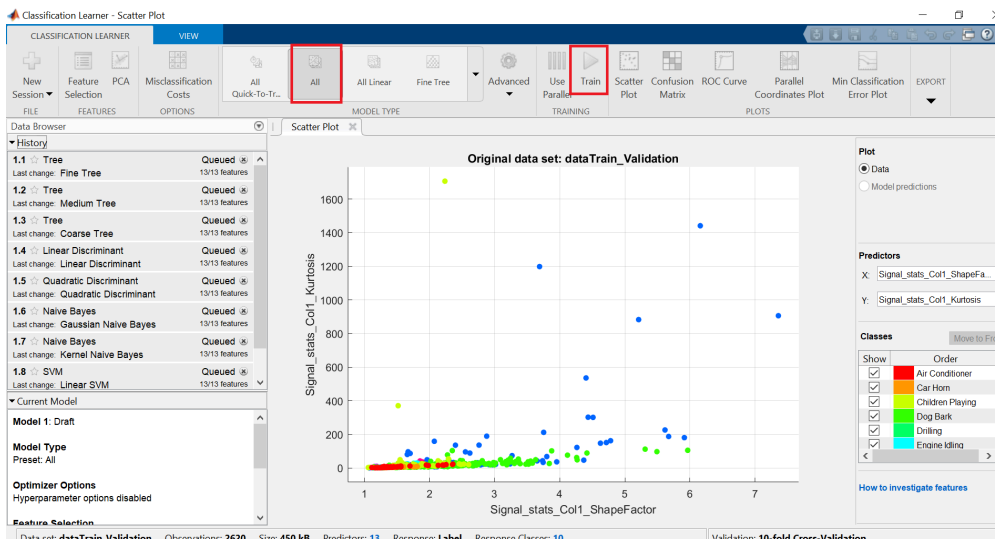
Percent held out: 25%

☐ **No Validation**

No protection against overfitting.

[Read about validation](#)

2) Train All Model



3) First run result, I found bag tree is the best model.

Data Browser

▼ History

Last change: **Boosted Trees**

1.21 ☆ Ensemble

Accuracy: **53.9%**

Last change: **Bagged Trees** 13/13 features

▼ Current Model

Total misclassification cost 1208

Prediction speed ~2500 obs/sec

Training time 17.152 sec

Model Type

Preset: Bagged Trees

Ensemble method: Bag

Learner type: Decision tree

Maximum number of splits: 2619

Number of learners: 30

Optimizer Options

Hyperparameter options disabled

Feature Selection

All features used in the model, before PCA

PCA

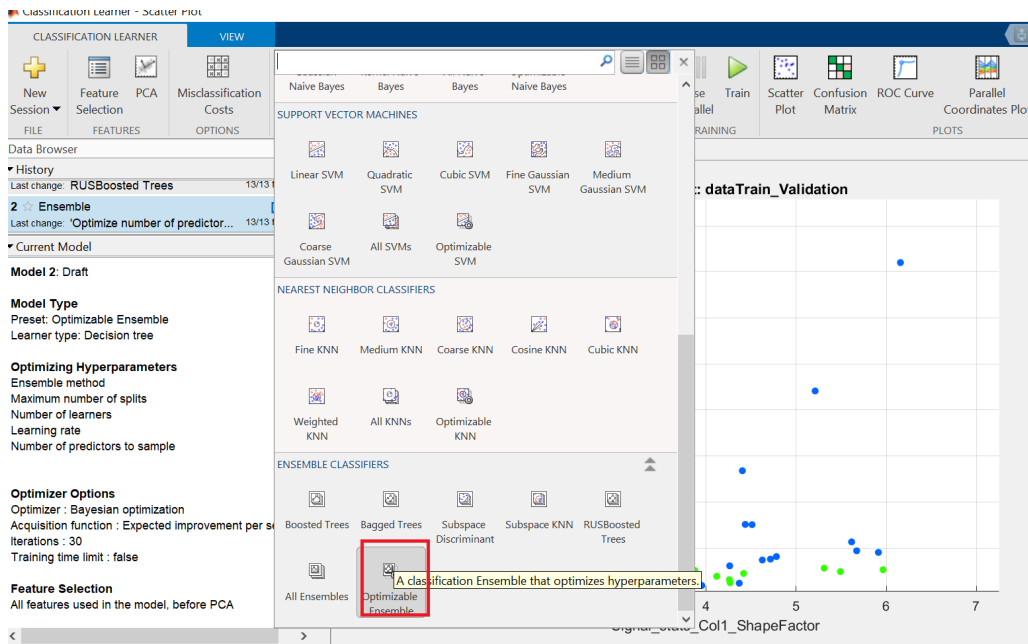
PCA disabled

Misclassification Costs

Cost matrix: default

4) Now, we try to use bayesian optimization to optimize the model. (play around with the hyperparameters, see whether can you get higher accuracy model).

Please take note, try to maintain the simplest model with high accuracy, else big model might cause us problem in code generation.



5) After Bayesian optimization, I still could not get model with better accuracy. Therefore, i try to impose higher misclassification costs on those wrong prediction more than 30.

		Model 3						
True class	Air Conditioner	206	1	14	4	22	19	
	Car Horn	10	32	6	5	34	10	
	Children Playing	15	1	128	31	32	11	3
	Dog Bark	11	1	32	194	21	5	4
	Drilling	14	9	14	8	182	4	
	Engine Idling	43	2	11	10	16	168	1
	Gun Shot	2		5	51	1	3	46
	Jackhammer	30		8	3	49	8	4
	Siren	13	6	28	6	12	9	
	Street Music	17	6	42	13	65	8	3
		Air Conditioner	Car Horn	Children Playing	Dog Bark	Drilling	Engine Idling	Gun Shot
		Predicted class						

		Predicted Class									
True Class		Air Conditioner	Car Horn	Children Playing	Dog Bark	Drilling	Engine Idling	Gun Shot	Jackhammer	Siren	Street Mus
	Air Conditioner	0	1	1	1	1	1	1	1	1	1
	Car Horn	1	0	1	1	1	1	1	1	1	1
	Children Playing	1	1	0	1	2	1	1	1	1	2
	Dog Bark	1	1	2	0	1	1	1	1	1	1
	Drilling	1	1	1	1	0	1	1	2	1	1
	Engine Idling	2	1	1	1	1	0	1	1	1	1
	Gun Shot	1	1	1	1	1	1	0	1	1	1
	Jackhammer	2	1	1	1	2	1	1	0	1	1
	Siren	1	1	1	1	1	1	1	1	0	0
	Street Music	1	1	2	1	1	1	1	1	1	2

6) After playing around the parameters/hyperparameters, the model below is the best. It has reasonable training time and prediction speed.

Last change: 'Cost matrix' = 'custom'
13/13 features

11 ☆ Ensemble
Accuracy: **54.9%**

Last change: 'Number of learners' = '100'
13/13 features

▼ Current Model

Model 11: Trained

Results

Accuracy 54.9%
Total misclassification cost 1471 (Models have different cost matrices)
Prediction speed ~1700 obs/sec
Training time 28.244 sec

Model Type

Preset: Bagged Trees
Ensemble method: Bag
Learner type: Decision tree
Maximum number of splits: 2619
Number of learners: 100

Optimizer Options

Hyperparameter options disabled

Feature Selection

All features used in the model, before PCA

Since i could not able to increase the accuracy significantly, as expected from visualizing the data before machine learning, we can't get model with good accuracy.

7) Export the model

Subspace Discriminant
Subspace KNN
Advanced
Use Parallel
Train
Scatter Plot
Confusion Matrix
ROC Curve
Parallel Coordinates Plot
Min Classification Error Plot
EXPORT

MODEL TYPE
TRAINING
PLOTS

Minimum Classification Error Plot
Confusion Matrix

Model 11

Air Conditioner
201
15
7
14
16

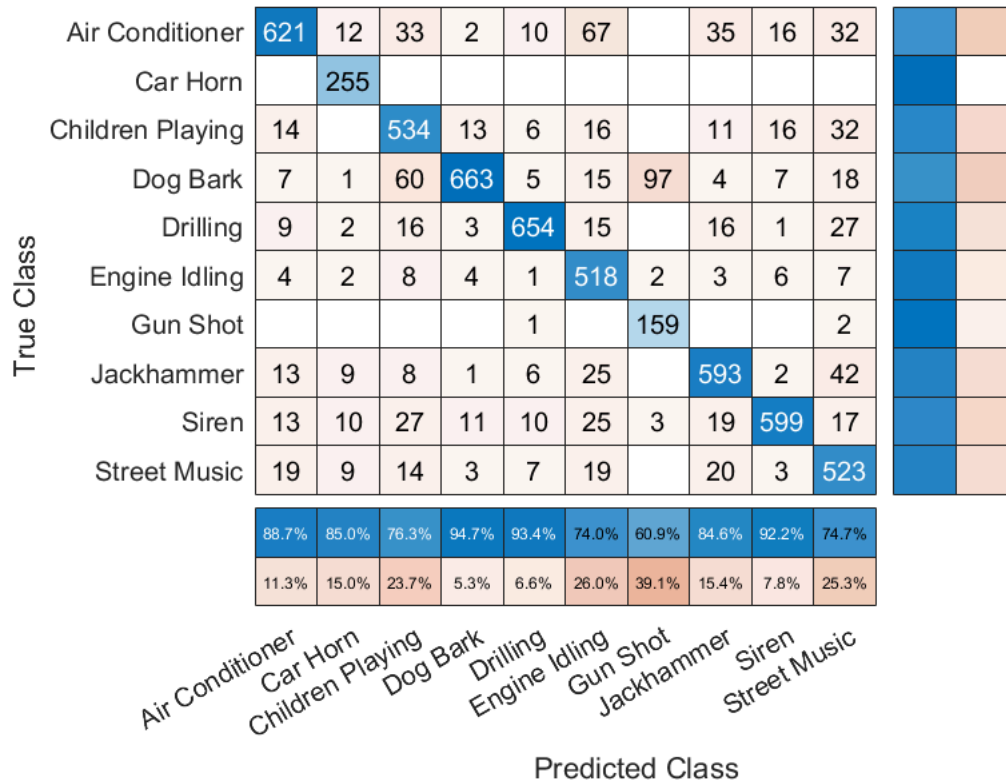
Export Plot to Figure
Generate Function
Export Model

Export Classification Model to Predict New Data

Make Predictions for New Data

After you export a model to the workspace from Classification Learner, or run the code generated from the app, you get a trainedModel structure that you can use to make predictions using new data. The structure contains a classification object and a function for prediction

```
[predictionoutcome,scores]=trainedModel.predictFcn(dataTest);
fig = figure;
cm = confusionchart(string(predictionoutcome),dataTest.Label,'RowSummary','row-normalized','Col
```



```
accuracy=sum(string(predictionoutcome)==dataTest.Label)/length(dataTest.Label)
```

```
accuracy = 0.8375
```

Accuracy is 83.75%

Observation :

We have noted down some important information before training, let review again :

Observation from ML1

1) With our eye, we can differentiate the sound using their shape of signal. Am I right? Now, we take note about it first.

- Shape factor is the most important factor in our ranking, therefore, we prove it right. However, with time domain and spectral features, we don't have much information about the shape, hence, we are only able to hit the prediction up to 55%.

Observation from ML2

1) if we look at the shape of signal, we will notice some are looking similar to each other although we still can differentiate it through sound. Especially, gun shot is looking similar to jackhammer. Drilling and Engine Idling are also looking similar.

- It is correct. In my intuitive, I try to impose higher penalty for the cost matrix to deal with this, however, I failed to increase the accuracy significantly.

2) if we are looking at the graph, many classes do not have significant distinction, we might not have good accuracy result if we only use time-domain and spectral features to train machine learning model.

- Now it is time for us to think what other features we can include which the features will tell more information about the shape. MFCC or Wavelet transform?

Generate MATLAB Code to Train the Model with New Data

After you create classification models interactively in Classification Learner, you can generate MATLAB code for your best model. You can then use the code to train the model with new data.

```
saveLearnerForCoder(trainedModel.ClassificationEnsemble, 'bagtree');
```

```
ans = 1x1 cell array
    {'Air Conditioner'}
```

```
function label = classifyurbansound(X) %#codegen
Mdl = loadLearnerForCoder('bagtree');
label = predict(Mdl,X);
end
```