Lecture 8

Dataset Partitioning and
Performance Evaluation Metrics

```
>> load fisheriris
>> X = meas;
>> Y = species;
>> Mdl = fitcnb(X,Y);
>> Mdl

Mdl =

  ClassificationNaiveBayes
            ResponseName: 'Y'
    CategoricalPredictors: []
              ClassNames: {1×3 cell}
          ScoreTransform: 'none'
         NumObservations: 150
       DistributionNames: {1×4 cell}
  DistributionParameters: {3×4 cell}


  Properties, Methods

>> doc resubLoss
>> L = resubLoss(Mdl)

L =

    0.0400
```

‘cvpartition_demo.m'

c = cvpartition(group,'KFold',k) creates a random partition for **stratified** k-fold cross-validation. Each subsample, or fold, has approximately the same number of observations and contains approximately the same class proportions as in group.

When you specify group as the first input argument, cvpartition discards rows of observations corresponding to missing values in group.

```
>> cv = cvpartition(C, 'KFold', 5);
>> C(cv.test(1))

ans =

     1     1     1     2     2     2

>> hist(C)
>> C(cv.test(2))

ans =

     1     1     1     2     2     2
```

Stratified sampling is a sampling technique where the samples are selected in the same proportion as they appear in the population, by dividing the population into groups called ‘strata’ based on a characteristic.

•

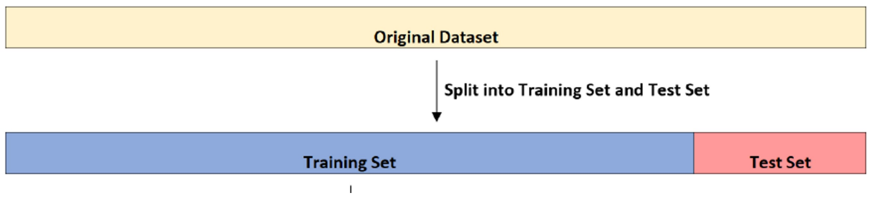
Implementing stratified sampling ensures the training and testing sets, or training and validation sets have the same proportion of the feature of interest as in the original dataset.

•

Thus the final trained model has a error performance that is a close approximation to the generalization error.

```
>> cv = cvpartition(C, 'KFold', 5, 'Stratify', false)

cv =

K-fold cross validation partition
   NumObservations: 30
       NumTestSets: 5
         TrainSize: 24  24  24  24  24
          TestSize: 6  6  6  6  6
>>
>> C(cv.test(1))

ans =

     1     1     2     2     2     2

>> C(cv.test(2))

ans =

     1     1     2     2     2     2
```

c = cvpartition(group,'Holdout',p) randomly partitions observations into a training set and a test, or holdout, set with stratification, using the class information in group. Both the training and test sets have approximately the same class proportions as in group.



```
>> cv = cvpartition(C, 'HoldOut', 0.2)

cv =

Hold-out cross validation partition
   NumObservations: 30
       NumTestSets: 1
         TrainSize: 24
          TestSize: 6

>> (cv.test)'

ans =

  1×30 logical array

  Columns 1 through 13

   0   1   0   0   0   0   0   0   0   0   0   1   0

  Columns 14 through 26

   0   1   1   1   0   1   0   0   0   0   0   0   0

  Columns 27 through 30

   0   0   0   0

>> cv = cvpartition(C, 'HoldOut', 0.2, 'Stratify', false)
>> C(cv.test)

ans =

     1     1     1     1     2     2
```

```
>> C(cv.test)

ans =

     1     1     1     2     2     2

X(cv.test)

>> X(cv.test)

ans =

  Columns 1 through 5

   -1.8188   -2.1997   -3.2428    3.1668    1.0346

  Column 6

    4.1752
```