h2omlestat aucmulticlass — Display AUC and AUCPR after multiclass classification[+]

[+]This command includes features that are part of StataNow.

Description

`h2omlestat aucmulticlass` reports area under the curve (AUC) and area under the precision–recall curve (AUCPR) metrics after multiclass classification performed by `h2oml gbmulticlass` or `h2oml rfmulticlass`. These metrics measure how well the model can classify observations. Unlike after binary classification, multiple variations of AUC and AUCPR metrics can be defined with multiclass classification. The variations include one-versus-one metrics, one-versus-rest metrics, and averages of these metrics.

AUC and AUCPR metrics can be computationally intensive. To obtain these metrics, the `auc` option must be specified in the `h2oml gbmulticlass` or `h2oml rfmulticlass` command before the metrics can be reported by `h2omlestat aucmulticlass`.

Quick start

Report AUC and AUCPR metrics

```
h2omlestat aucmulticlass
```

As above, but report testing results based on data in frame `test`

```
h2omlestat aucmulticlass, test(test)
```

Menu

Statistics > H2O machine learning

Syntax

```
h2omlestat aucmulticlass [ , options ]
```

options	Description
title(*string*)	specify title to be displayed above the table
train	specify that metrics be reported using training results
valid	specify that metrics be reported using validation results
cv	specify that metrics be reported using cross-validation results
test	specify that metrics be computed using the testing frame
test(*framename*)	specify that metrics be computed using data in testing frame *framename*
frame(*framename*)	specify that metrics be computed using data in H2O frame *framename*
framelabel(*string*)	label frame as *string* in the output

collect is allowed; see **[U] 11.1.10 Prefix commands**.

train, valid, cv, test, test(), frame(), and framelabel() do not appear in the dialog box.

Options

title(*string*) specifies the title to be displayed above the table.

The following options are available with h2omlestat aucmulticlass but are not shown in the dialog box:

train, valid, cv, test, test(), and frame() specify the H2O frame for which AUC and AUCPR metrics are reported. Only one of train, valid, cv, test, test(), or frame() is allowed.

train specifies that AUC and AUCPR metrics be reported using training results. This is the default when neither validation nor cross-validation is performed during estimation and when a postestimation frame has not been set with h2omlpostestframe.

valid specifies that AUC and AUCPR metrics be reported using validation results. This is the default when validation is performed during estimation and when a postestimation frame has not been set with h2omlpostestframe. valid may be specified only when the validframe() option is specified with h2oml *gbm* or h2oml *rf*.

cv specifies that AUC and AUCPR metrics be reported using cross-validation results. This is the default when cross-validation is performed during estimation and when a postestimation frame has not been set with h2omlpostestframe. cv may be specified only when the cv or cv() option is specified with h2oml *gbm* or h2oml *rf*.

test specifies that AUC and AUCPR metrics be computed on the testing frame specified with h2omlpostestframe. This is the default when a testing frame is specified with h2omlpostestframe. test may be specified only after a testing frame is set with h2omlpostestframe. test is necessary only when a subsequent h2omlpostestframe command is used to set a default postestimation frame other than the testing frame.

`test(`*framename*`)` specifies that AUC and AUCPR metrics be computed using data in testing frame *framename* and is rarely used. This option is most useful when running a single postestimation command on the named frame. If multiple postestimation commands are to be run on the same test frame, `h2omlpostestframe` provides a more convenient and computationally efficient process for doing this.

`frame(`*framename*`)` specifies that AUC and AUCPR metrics be computed using the data in H2O frame *framename*.

`framelabel(`*string*`)` specifies the label to be used for the frame in the output. This option is not allowed with the `cv` option.

stata.com

Remarks and examples

`h2omlestat aucmulticlass` computes AUC and AUCPR metrics after multiclass classification. These metrics measure how well the model can classify observations. Unlike with binary classification, observations are not classified into simply one positive and one negative class. Instead, with multiclass classification, variations of these metrics are defined. The one-versus-one metrics compute the AUC and AUCPR for all pairwise combinations of the classes. The one-versus-rest metrics compute the AUC and AUCPR for each class versus all the other classes combined. `h2omlestat aucmulticlass` reports all one-versus-one and one-versus-rest AUC and AUCPR metrics. It also reports the macro (unweighted) average and the prevalence weighted average of each metric. For definitions of these metrics, see [H2OML] ***metric_option***.

Because calculation of the AUC and AUCPR metrics is computationally expensive for multiclass classification, these metrics are not calculated by default by `h2oml gbmulticlass` and `h2oml rfmulticlass`. To enable the calculation, we must specify the `auc` option during estimation. Additionally, AUC and AUCPR metrics may not be requested when the number of response classes is greater than 50.

▷ Example 1: AUC and AUCPR metrics

We use a well-known `iris` dataset, where the goal is to predict a class of iris plant. This dataset was used in Fisher (1936) and originally collected by Anderson (1935). We start by initializing a cluster, opening the dataset in Stata, and importing the dataset as an H2O frame. Recall that `h2o init` initiates an H2O cluster, `_h2oframe put` loads the current Stata dataset into an H2O frame, and `_h2oframe change` makes the specified frame the current H2O frame. For details, see *Prepare your data for H2O machine learning in Stata* in [H2OML] **h2oml** and see [H2OML] **H2O setup**.

```
. use https://www.stata-press.com/data/r18/iris
(Iris data)
. h2o init
. _h2oframe put, into(iris)
. _h2oframe change iris
```

We define the global macro `predictors` to store the names of the predictors, and we use the `h2oml rfmulticlass` command to perform random forest multiclass classification. We use default settings for all hyperparameters, and we specify an H2O random-number seed for reproducibility. We also specify the `auc` option to request that the AUC and AUCPR metrics be computed.

```
. global predictors seplen sepwid petlen petwid
. h2oml rfmulticlass iris $predictors, h2orseed(19) auc
Progress (%): 0 100
Random forest multiclass classification using H2O
Response: iris                          Number of classes    =      3
Frame:                                  Number of observations:
  Training: iris                                  Training =    150
Model parameters
Number of trees      =  50
             actual =  50
Tree depth:                             Pred. sampling value =     -1
           Input max =  20              Sampling rate        =   .632
                 min =   1              No. of bins cat.     =  1,024
                 avg = 3.7              No. of bins root     =  1,024
                 max =   9              No. of bins cont.    =     20
Min. obs. leaf split =   1              Min. split thresh.   = .00001
Metric summary
------------------------------------------
           Metric |    Training
------------------------------------------
         Log loss |    .3438683
 Mean class error |    .0533333
              AUC |    .9906667
            AUCPR |    .9816699
              MSE |    .0384685
             RMSE |     .196134
------------------------------------------
Note: AUC and AUCPR computed
      using macro average OVR.
```

The output reports an AUC of 0.991 and an AUCPR of 0.982. The note at the bottom of the table tells us that these values are the macro average OVR (one-versus-rest) metrics.

To report all computed AUC and AUCPR metrics, we type

```
. h2omlestat aucmulticlass
AUC and AUCPR summary using H2O
Training frame: iris
```

	AUC	AUCPR
One vs. rest (OVR)		
Setosa vs. rest	1	1
Versicolor vs. rest	.983	.978
Virginica vs. rest	.989	.967
Macro OVR	.991	.982
Weighted OVR	.991	.982
One vs. one (OVO)		
Setosa vs. Versicolor	.995	.997
Setosa vs. Virginica	1	1
Versicolor vs. Virginica	.977	.974
Macro OVO	.991	.99
Weighted OVO	.991	.99

As with standard AUC, a value closer to 1 for each of these metrics indicates better classification. In the first table, we see the one-versus-rest AUC values followed by the one-versus-one AUC values. The `Setosa vs. Rest` AUC value is 1. This means that if we run a binary classification where `Setosa` is considered the positive class and the remaining classes are considered the negative class, then the model will perfectly classify all observations.

Similarly, the `Versicolor vs. Rest` AUC is the AUC for a binary classification where `Versicolor` is treated as the positive class and the other classes jointly comprise the negative class. `Macro OVR` is an unweighted average of the above one-versus-rest AUCs that gives all classes the same weight. `Weighted OVR` is a prevalence weighted average of the one-versus-rest AUCs, where weights are assigned to classes based on the number of positives in each class.

In the next portion of the first table, the AUCs are computed by treating one class as the positive class and one class as the negative class while ignoring all other classes.

The second table can be interpreted similarly to the first table, but it reports AUCPR metrics rather than AUC metrics. The AUCPR is preferred when the classes of the response variable are highly imbalanced.

In this example, all the reported AUC and AUCPR metrics are close to 1, indicating that the model can accurately distinguish between each class and the other classes. However, as we illustrate in the next example, this does not mean that the model is highly accurate at performing multiclass classification in terms of assigning the correct class to every observation.

◁

Example 2: AUC and AUCPR for validation and testing frames

Above, we performed classification and evaluated metrics using a single training frame. To demonstrate how to obtain the AUC and AUCPR metrics for other frames, such as validation and testing frames, we first use the _h2oframe _split command to split the dataset, specifying 60% of observations in the training frame, 20% in the validation frame, and 20% in the testing frame. We then change to the training frame.

```
. use https://www.stata-press.com/data/r18/iris, clear
(Iris data)
. h2o init
. _h2oframe put, into(iris)
. _h2oframe split iris, into(training validation testing) split(0.6 0.2 0.2)
> rseed(19)
. _h2oframe change training
```

Next we perform random forest multiclass classification, setting the number of trees to 500 and leaving the other hyperparameters at their default values. We also specify the name of our validation frame in the validframe() option.

```
. h2oml rfmulticlass iris $predictors, h2orseed(19) auc ntrees(500)
> validframe(validation)
Progress (%): 0 29.6 53.3 71.3 99.1 100
Random forest multiclass classification using H2O
Response: iris                          Number of classes    =      3
Frame:                                  Number of observations:
  Training:   training                                Training =     95
  Validation: validation                            Validation =     30
Model parameters
Number of trees      = 500
             actual = 500
Tree depth:                             Pred. sampling value =     -1
           Input max =  20              Sampling rate        =   .632
                 min =   1              No. of bins cat.     =  1,024
                 avg = 3.0              No. of bins root     =  1,024
                 max =   9              No. of bins cont.    =     20
Min. obs. leaf split =   1              Min. split thresh.   = .00001
Metric summary
```

Metric	Training	Validation
Log loss	.1027022	.1406913
Mean class error	.0423591	.0666667
AUC	.995535	1
AUCPR	.9915411	1
MSE	.0300273	.0473201
RMSE	.1732838	.2175318

```
Note: AUC and AUCPR computed using macro
      average OVR.
```

Now we can run h2omlestat aucmulticlass to see how well our model classifies the data in the validation frame. Because we specified the validation frame during estimation, h2omlestat aucmulticlass defaults to reporting metrics for the validation frame.

```
. h2omlestat aucmulticlass
AUC and AUCPR summary using H2O
Validation frame: validation
```

	AUC	AUCPR
One vs. rest (OVR)		
Setosa vs. rest	1	1
Versicolor vs. rest	1	1
Virginica vs. rest	1	1
Macro OVR	1	1
Weighted OVR	1	1
One vs. one (OVO)		
Setosa vs. Versicolor	1	1
Setosa vs. Virginica	1	1
Versicolor vs. Virginica	1	1
Macro OVO	1	1
Weighted OVO	1	1

We get a score of 1 for each of the one-versus-rest AUC metrics, meaning that if we performed three binary classifications, one for each class being positive while the rest of the classes are negative, those models will correctly classify all observations. Similarly, all the one-versus-one AUC metrics are 1, corresponding to perfect prediction for all pairwise binary classifications where one class is considered positive and another is considered negative.

However, it is important to remember that computation of one-versus-one AUC and one-versus-rest AUC metrics ignores the fact that the initial problem is multiclass. The results can differ compared with other performance metrics that take into account the true multiclass nature of the problem. For example, let's look at the confusion matrix by using the h2omlestat confmatrix command.

```
. h2omlestat confmatrix
Confusion matrix using H2O
Validation frame: validation
```

	Predicted					
iris	Setosa	Versico~r	Virginica	Total	Error	Rate
Setosa	12	0	0	12	0	0
Versicolor	0	8	0	8	0	0
Virginica	0	2	8	10	2	.2
Total	12	10	8	30	2	.067

We see that Setosa and Versicolor were perfectly classified, but the model did misclassify some Virginica flowers as Versicolor.

In addition to the default metrics that are reported for the validation frame in this case, we can obtain metrics for other frames. Here we are interested in results from the testing frame, and we have two ways to request these. One approach is to use the test(testing) option to specify the testing frame. The second approach, our preferred method, is to use h2omlpostestframe to set the testing frame to be used as the default for all affected postestimation commands. For details, see [H2OML] **h2omlpostestframe**.

```
. h2omlpostestframe testing
(testing frame testing is now active for h2oml postestimation)
. h2omlestat aucmulticlass
AUC and AUCPR summary using H2O
Testing frame: testing
```

	AUC	AUCPR
One vs. rest (OVR)		
Setosa vs. rest	1	1
Versicolor vs. rest	1	1
Virginica vs. rest	1	1
Macro OVR	1	1
Weighted OVR	1	1
One vs. one (OVO)		
Setosa vs. Versicolor	1	1
Setosa vs. Virginica	1	1
Versicolor vs. Virginica	1	1
Macro OVO	1	1
Weighted OVO	1	1

As with the validation frame, we obtain values of 1 for all AUC and AUCPR metrics calculated on the testing frame.

◁

Stored results

h2omlestat aucmulticlass stores the following in r():

Matrices

r(aucmulticlass)	one-versus-rest and one-versus-one AUC scores
r(aucprmulticlass)	one-versus-rest and one-versus-one AUCPR scores

References

Anderson, E. 1935. The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society* 59: 2–5.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7: 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Also see

[H2OML] **h2oml** — Introduction to commands for Stata integration with H2O machine learning+

[H2OML] **h2omlestat confmatrix** — Display confusion matrix+