h2oml postestimation — Postestimation tools for h2oml gbm and h2oml rf

Postestimation commands   h2omlpredict   Remarks and examples   References
Also see

Postestimation commands

The following postestimation commands are of special interest after `h2oml` *gbm* and `h2oml` *rf*:

Command	Description
Estimation results and postestimation frame	
`h2omlest`	store and restore estimation results
`h2omlpostestframe`	specify frame for postestimation analysis
Tuning and estimation summaries	
`h2omlestat metrics`	display performance metrics
`h2omlgraph scorehistory`	produce score history plot
`h2omlestat cvsummary`	display cross-validation summary
`h2omlestat gridsummary`	display grid-search summary
`h2omlexplore`	explore models after grid search
`h2omlselect`	select model after grid search
`h2omlgof`	compare goodness of fit for machine learning models
Model performance after binary classification	
`h2omlestat threshmetric`	display threshold-based metrics
`h2omlgraph prcurve`	produce precision–recall curve plot
`h2omlgraph roc`	produce ROC curve plot
Model performance after multiclass classification	
`h2omlestat aucmulticlass`	display AUC and AUCPR metrics
`h2omlestat hitratio`	display hit-ratio table
Model performance after binary and multiclass classification	
`h2omlestat confmatrix`	display confusion matrix
Prediction	
`h2omlpredict`	predict continuous responses, probabilities, and classes
Model explainability	
`h2omlgraph varimp`	produce variable importance plot
`h2omlgraph pdp`	produce partial dependence plot
`h2omlgraph ice`	produce individual conditional expectation plot
`h2omltree`	save decision tree DOT file and display rule set
Explainability after regression and binary classification	
`h2omlgraph shapvalues`	produce SHAP values plot for individual observations
`h2omlgraph shapsummary`	produce SHAP beeswarm plot

h2omlpredict

Description for h2omlpredict

`h2omlpredict` generates new variables (H2O columns) containing predictions, probabilities, and class predictions. The latter two are provided for the binary and multiclass classification problems.

Menu for h2omlpredict

Statistics > H2O machine learning

Syntax for h2omlpredict

After `h2oml gbregress` and `h2oml rfregress`

`h2omlpredict` *newvar* [, `frame(`*framename*`)`]

After `h2oml gbbinclass` and `h2oml rfbinclass`

`h2omlpredict` *stub*`*` | *newvar* | *newvarlist* [, *binopts* `frame(`*framename*`)`]

After `h2oml gbmulticlass` and `h2oml rfmulticlass`

`h2omlpredict` *stub*`*` | *newvar* | *newvarlist* [, *multopts* `frame(`*framename*`)`]

binopts	Description
Main	
`class`	predicted classes
`pr`	predicted probability of each class
`threshold(#)`	specify threshold for predicting classes

multopts	Description
Main	
`class`	predicted classes
`pr`	predicted probability of each class
`outcome(`*outcome*`)`	specify outcome level (class) for which probabilities are computed

You specify one or k new variables with `pr`, where k is the number of outcomes. If you specify one new variable and you do not specify `outcome()`, then `outcome(#1)` is assumed.

Options for h2omlpredict

Main

`frame(`*framename*`)` specifies the H2O frame in which predictions are stored.

`class` computes class predictions for each observation and is the default. For `h2oml gbbinclass` and `h2oml rfbinclass`, the predicted class for each observation is determined based on a threshold value. By default, the threshold is set to maximize the F1 score. Alternatively, a custom threshold can be specified using the `threshold()` option. For `h2oml gbmulticlass` and `h2oml rfmulticlass`, the predicted class for each observation is based on the highest predicted probability. Only one of `class` or `pr` is allowed.

`pr` computes the predicted probabilities for all outcome levels (classes) or for a specific outcome level (class) after classification. To compute probabilities for all outcome levels, you specify k new variables (H2O columns), where k is the number of classes of the response. Alternatively, you can specify *stub*`*`, in which case `pr` will store predicted probabilities in variables (H2O columns) *stub1*, *stub2*, ..., *stubk*. To compute the probability for a specific outcome level, you specify one new variable (H2O column) and, optionally, the outcome value in option `outcome()`; if you omit `outcome()`, then the first outcome value, `outcome(#1)`, is assumed. Say that you fit a model by typing `h2oml` *estimation_cmd* `y x1 x2`, and `y` has four classes. Then you could type `h2omlpredict p1 p2 p3 p4, pr` to obtain all four predicted probabilities; alternatively, you could type `h2omlpredict p*, pr` to generate the four predicted probabilities. To compute specific probabilities one at a time, you can type `h2omlpredict p1, pr outcome(#1)` (or simply `h2omlpredict p1, pr`); `h2omlpredict p2, pr outcome(#2)`; and so on. See the `outcome()` option for other ways to refer to the outcome value. Only one of `pr` or `class` is allowed.

`threshold(#)` specifies the threshold for predicted classes for binary classification. The specified number should be between $[0, 1]$. By default, the threshold value that maximizes the F1 metric is used.

`outcome(`*outcome*`)` specifies for which outcome level (class) the predicted probabilities are to be calculated after multiclass classification. `outcome()` should contain either one class of the response or one of `#1`, `#2`, ..., with `#1` meaning the first class of the response, `#2` meaning the second class, etc. `outcome()` is not allowed with `class`.

Remarks and examples

Remarks and examples are presented under the following headings:

Binary classification prediction

▷ Example 1

In this example, we show how to use the `h2omlpredict` command to predict probabilities and classes for binary classification.

We start by opening the 1978 automobile data (`auto.dta`) in Stata and then putting the data into 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/r19/auto
(1978 automobile data)

. h2o init
  (output omitted)

. _h2oframe put, into(auto)
Progress (%): 0 100

. _h2oframe change auto
```

We use h2oml rfbinclass to perform random forest binary classification to predict classes of the car origin.

```
. global predictors price mpg length weight

. h2oml rfbinclass foreign $predictors, ntrees(100) h2orseed(19)
Progress (%): 0 100

Random forest binary classification using H2O

Response: foreign
Frame:                                 Number of observations:
  Training: auto                                   Training =     74

Model parameters

Number of trees       = 100
              actual = 100
Tree depth:                            Pred. sampling value =     -1
           Input max =  20             Sampling rate        =   .632
                 min =   3             No. of bins cat.     =  1,024
                 avg = 5.5             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   .3053323
       Mean class error    .1284965
                     AUC   .9309441
                  AUCPR    .8455917
        Gini coefficient   .8618881
                     MSE   .1046538
                    RMSE   .3235024
```

Next we use h2omlpredict to create a new variable (a column in the current H2O frame) containing the predicted classes.

```
. h2omlpredict foreignhat, class
Progress (%): 0 100
```

The threshold value is a cutpoint that determines the predicted classes from the predicted probabilities. In binary classification, the threshold is the value that maximizes the F1 score. We can determine this threshold value by using `h2omlestat threshmetric`.

```
. h2omlestat threshmetric

Maximum or minimum metrics using H2O
Training frame: auto
```

Metric	Max/Min	Threshold
F1	.7778	.125
F2	.8871	.0732
F0.5	.7979	.6286
Accuracy	.8649	.6286
Precision	1	1
Recall	1	.0732
Specificity	1	1
Min. class accuracy	.8269	.2258
Mean class accuracy	.8715	.125
True negatives	52	1
False negatives	0	.0732 +
True positives	22	.0732
False positives	0	1 +
True-negative rate	1	1
False-negative rate	0	.0732 +
True-positive rate	1	.0732
False-positive rate	0	1 +
MCC	.6855	.125

```
+ identifies minimum metrics.
```

The threshold that maximizes the F1 score is 0.125. Thus, the observations with predicted probabilities greater than 0.125 are assigned to the positive class (`Foreign` in our example), and the remaining observations are assigned to the negative class (`Domestic` in our example). We can specify a different threshold with the `threshold()` option. For example, we can select the threshold that maximizes the true-positive rate, which is 0.0732.

```
. h2omlpredict foreignhat_tpr, class threshold(0.0732)
```

If we want to obtain predicted probabilities, we can use the `pr` option.

```
. h2omlpredict foreignpr1 foreignpr2, pr
Progress (%): 0 100
```

We can get the predictions and the rest of the data in the H2O frame back into Stata by using the `_h2oframe get` command.

```
. clear
. _h2oframe get auto
```

◁

Multiclass classification prediction

▷ Example 2

In this example, we show how to use the h2omlpredict command to predict probabilities and classes for multiclass classification.

For this example, we will 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. We then use the _h2oframe split command to randomly split the iris frame into a training frame (80% of observations) and a testing frame (20% of observations), which we name train and test, respectively. We also change the current frame to train.

```
. use https://www.stata-press.com/data/r19/iris
(Iris data)

. h2o init
  (output omitted)

. _h2oframe put, into(iris)

Progress (%): 0 100

. _h2oframe split iris, into(train test) split(0.8 0.2) rseed(19)

. _h2oframe change train
```

Next, we use h2oml rfmulticlass to perform random forest multiclass classification.

```
. global predictors seplen sepwid petlen petwid

. h2oml rfmulticlass iris $predictors, ntrees(100) h2orseed(19)

Progress (%): 0 100

Random forest multiclass classification using H2O

Response: iris                          Number of classes     =      3
Frame:                                  Number of observations:
  Training: train                                  Training =    125

Model parameters

Number of trees       = 100
             actual = 100
Tree depth:                             Pred. sampling value =     -1
          Input max =  20               Sampling rate        =   .632
                min =   1               No. of bins cat.     =  1,024
                avg = 3.5               No. of bins root     =  1,024
                max =   8               No. of bins cont.    =     20
Min. obs. leaf split =   1              Min. split thresh.   = .00001

Metric summary
```

Metric	Training
Log loss	.1282741
Mean class error	.0650407
MSE	.0389344
RMSE	.197318

Now, we use h2omlpredict to obtain the predicted classes of the iris plant.

```
. h2omlpredict irishat, class

Progress (%): 0 100
```

For multiclass classification, the class is assigned based on the class with the largest predicted probability. We can use the pr option to see the predicted probabilities. The number of specified new variable names should correspond to the number of classes (or we can specify *stub**, such as irispr*).

```
. h2omlpredict irispr1 irispr2 irispr3, pr
Progress (%): 0 100
```

By default, the variables (H2O columns) corresponding to the predicted probabilities and classes are created in the current frame, which in our case is train.

◁

Testing frame prediction

▷ Example 3

We continue the previous example and show how to obtain predictions on the testing data. In general, there are two approaches to achieve this goal.

In the first approach, which we recommend, we use the h2omlpostestframe command.

```
. h2omlpostestframe test
(testing frame test is now active for h2oml postestimation)
. h2omlpredict irishat, class
Progress (%): 0 100
```

The above commands generate variable irishat in the frame test.

In the second approach, we use the frame() option.

```
. h2omlpredict irishat1, class frame(test)
```

Note that neither approach physically changes the working frame to the specified frame, test.

If we are interested in listing the generated variable, then we can type the following.

```
. _h2oframe change test
. _h2oframe list in 1/5
     iris  seplen  sepwid  petlen  petwid  irishat  irishat1
1 Setosa      4.7     3.2     1.3      .2   Setosa    Setosa
2 Setosa      5.1     3.8     1.5      .3   Setosa    Setosa
3 Setosa      5.1     3.7     1.5      .4   Setosa    Setosa
4 Setosa      5.5     4.2     1.4      .2   Setosa    Setosa
5 Setosa      4.9     3.6     1.4      .1   Setosa    Setosa
[5 rows x 7 columns]
```

◁

Regression prediction

▷ Example 4

In this example, we show how to obtain predictions for regression.

We again use `auto.dta`.

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)
. h2o init
  (output omitted)
. _h2oframe put, into(auto)
Progress (%): 0 100
. _h2oframe change auto
```

We perform gradient boosting regression to predict prices.

```
. h2oml gbregress price mpg weight length, ntrees(100) h2orseed(19)
Progress (%): 0 100
Gradient boosting regression using H2O
Response: price
Loss:     Gaussian
Frame:                                     Number of observations:
  Training: auto                                      Training =     74
Model parameters
Number of trees       = 100                Learning rate       =     .1
              actual  = 100                Learning rate decay =      1
Tree depth:                                Pred. sampling rate =      1
           Input max  =   5                Sampling rate       =      1
                 min  =   3                No. of bins cat.    =  1,024
                 avg  = 4.1                No. of bins root    =  1,024
                 max  =   5                No. of bins cont.   =     20
Min. obs. leaf split  =  10                Min. split thresh.  = .00001
Metric summary
```

Metric	Training
Deviance	1612524
MSE	1612524
RMSE	1269.852
RMSLE	.1750365
MAE	853.3532
R-squared	.8121031

Then we use `h2omlpredict` to obtain predictions.

```
. h2omlpredict pricehat
Progress (%): 0 100
```

The new variable (H2O column) `pricehat` now contains the predicted prices based on our model.

◁

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

For suggested citations, see the FAQ on citing Stata documentation.