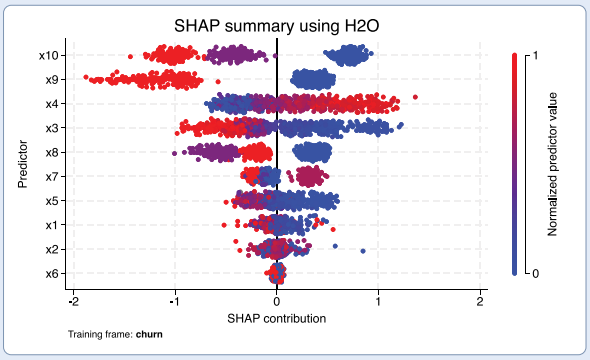
H2O machine learning

Ensemble decision trees

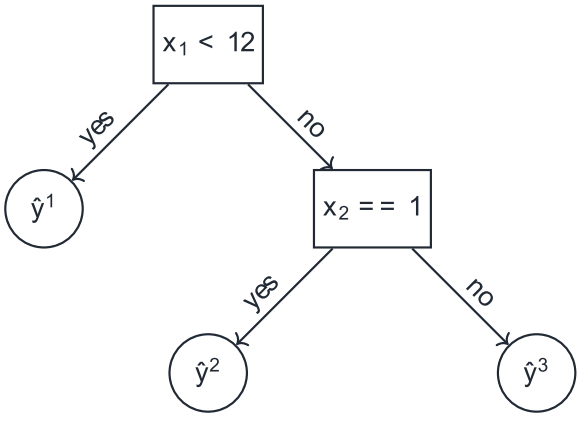
The **h2oml** command combines Stata's easy syntax with H2O's powerful machine learning tools.

- Random forest
- Gradient boosting machine
- Regression and classification
- Continuous, count, binary, and categorical responses
- Cross-validation and grid-search summary
- Hyperparameter tuning, model performance, and prediction
- Prediction explainability, including SHAP values and partial dependence plots
- More



Machine learning (ML) provides various statistical methods that answer complex, scientific, predictive questions about a response of interest based on the observed predictors. For example, given their credit history, how likely will borrowers default on a loan? Or how much may house prices change given a 5% property tax increase? Ensemble decision trees are a popular ML method for answering such questions.

A decision tree is a result of partitioning predictors' values into nonoverlapping regions such that the errors of incorrectly predicting responses in all of these regions are as small as possible. Ensemble decision trees, such as random forest and gradient boosting machine, combine multiple decision trees to improve prediction accuracy.



H2O setup in Stata

To use the **h2oml** command, we must first start a new H2O cluster or connect to an existing H2O cluster from within Stata and prepare an H2O data frame:

```
. h2o init
. _h2oframe put, into(mydata) current
```

Random forest

Random forest linear regression

```
. h2oml rfregress y1 x1-x10, ...
```

Random forest binary classification

```
. h2oml rfbinclass y2 x1-x10, ...
```

Random forest multiclass classification

```
. h2oml rfmulticlass y3 x1-x10, ...
```

Gradient boosting machine

Gradient boosting linear regression

```
. h2oml gbregress y1 x1-x10, ...
```

Gradient boosting binary classification

```
. h2oml gbbinclass y2 x1-x10, ...
```

Gradient boosting multiclass classification

```
. h2oml gbmulticlass y3 x1-x10, ...
```

Gradient boosting Poisson regression

```
. h2oml gbregress y4 x1-x10, loss(poisson) ...
```

Gradient boosting quantile regression with monotonicity constraint

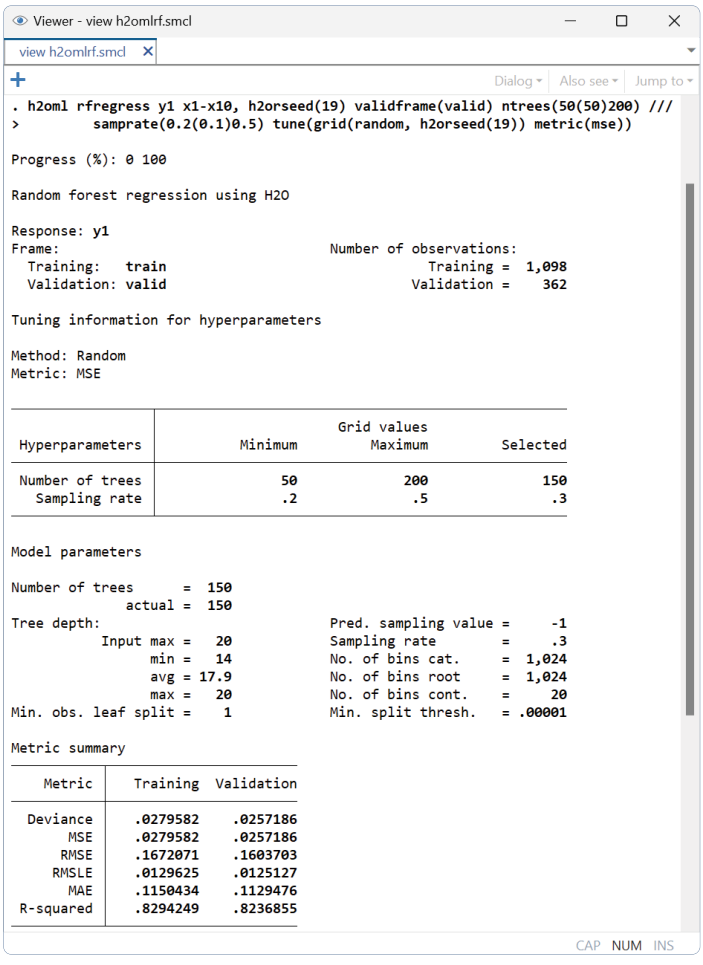
```
. h2oml gbregress y1 x1-x10, loss(quantile)
        monotone(x1, increasing) ...
```

Random forest with hyperparameter tuning and prediction explainability

Perform random forest linear regression of **y1** on **x1** through **x10** with validation

```
. h2oml rfregress y1 x1-x10, h2orseed(19)
        validframe(valid)
```

Perform random forest linear regression with hyperparameter tuning for the number of trees and sampling rate



Explore grid summary

```
. h2omlestat gridsummary
```

Produce variable importance plot

```
. h2omlgraph varimp
```

Produce partial dependence plots for the important predictors

```
. h2omlgraph pdp x1 x2 x5
```

Produce SHAP summary plot for top 10 SHAP important predictors

```
. h2omlgraph shapsummary
```

Gradient boosting model performance and prediction

Perform gradient boosting binary classification of **y2** on **x1** through **x10** with 3-fold stratified cross-validation

```
. h2oml gbbinclass y2 x1-x10, h2orseed(19)
        cv(3, stratify)
```

Perform gradient boosting binary classification, and set the number of trees to 30 and the maximum tree depth to 10

```
. h2oml gbbinclass y2 x1-x10, h2orseed(19)
        cv(3, stratify) ntrees(30) maxdepth(10)
```

Explore cross-validation summary

```
. h2omlestat cvsummary
```

Summarize classification prediction using confusion matrix

```
. h2omlestat confmatrix
```

Plot ROC and precision–recall curves to evaluate the model's performance

```
. h2omlgraph roc
. h2omlgraph prcurve
```

Predict classes and their probabilities using testing dataset

```
. h2omlpostestframe test
. h2omlpredict hatclass, class
. h2omlpredict phat1 phat0, pr
```