Politecnico di Bari

Dipartimento di Ingegneria Elettrica e dell'Informazione

Corso di Laurea Triennale in Ingegneria dei Sistemi Medicali

Bioinformatics and Big Data Analytics

Decision Trees

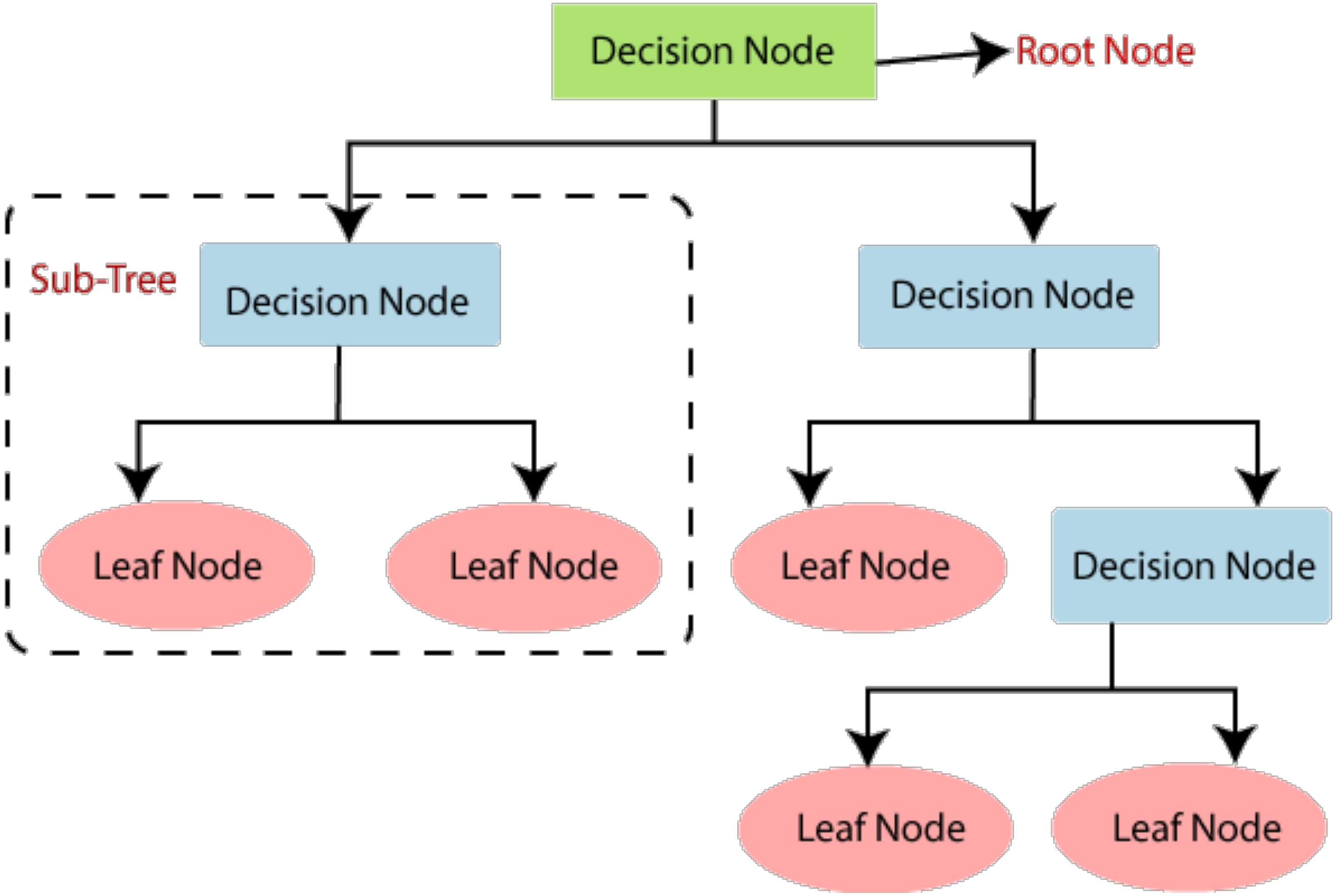
Eng. Nicola **Altini**, *Ph.D. Student*
Eng. Giacomo Donato **Cascarano**, *Ph.D. Student*
Prof. *Eng*. Vitoantonio **Bevilacqua**, *Ph.D.*

Anno Accademico 2019/2020

Decision Tree

- A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
- A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification (or regression) rules.

Decision Tree



C4.5 algorithm

- C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.
- J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool.
- C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = \{s^{(1)}, s^{(2)}, \ldots, s^{(n)}\}$ of already classified samples.
- Each sample $s^{(i)}$ consists of a m-dimensional vector $(x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)})$, where the x_j represent attribute values or features of the sample, as well as the class in which $s^{(i)}$ falls.

C4.5 algorithm

- At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the partitioned sublists.
- This algorithm has a few base cases.
  - All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
  - None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
  - Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

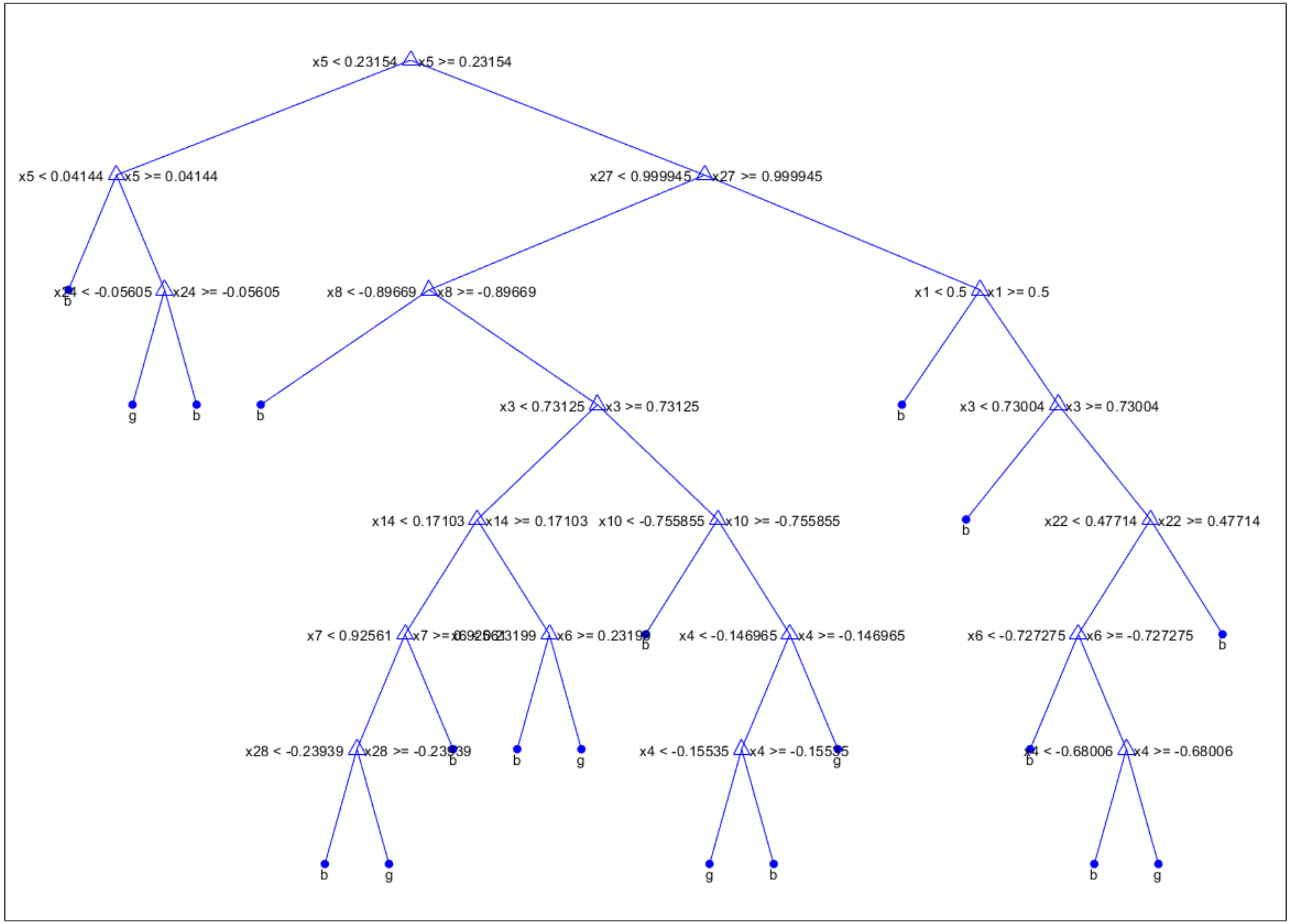
Classification Tree in MATLAB

- **Class:** **`ClassificationTree`**
- Superclasses: `CompactClassificationTree`
- Binary decision tree for classification
- Description
  - A `ClassificationTree` object represents a decision tree with binary splits for classification. An object of this class can predict responses for new data using the predict method. The object contains the data used for training, so it can also compute resubstitution predictions.
- Construction
  - Create a `ClassificationTree` object by using `fitctree`.

Classification Tree in MATLAB

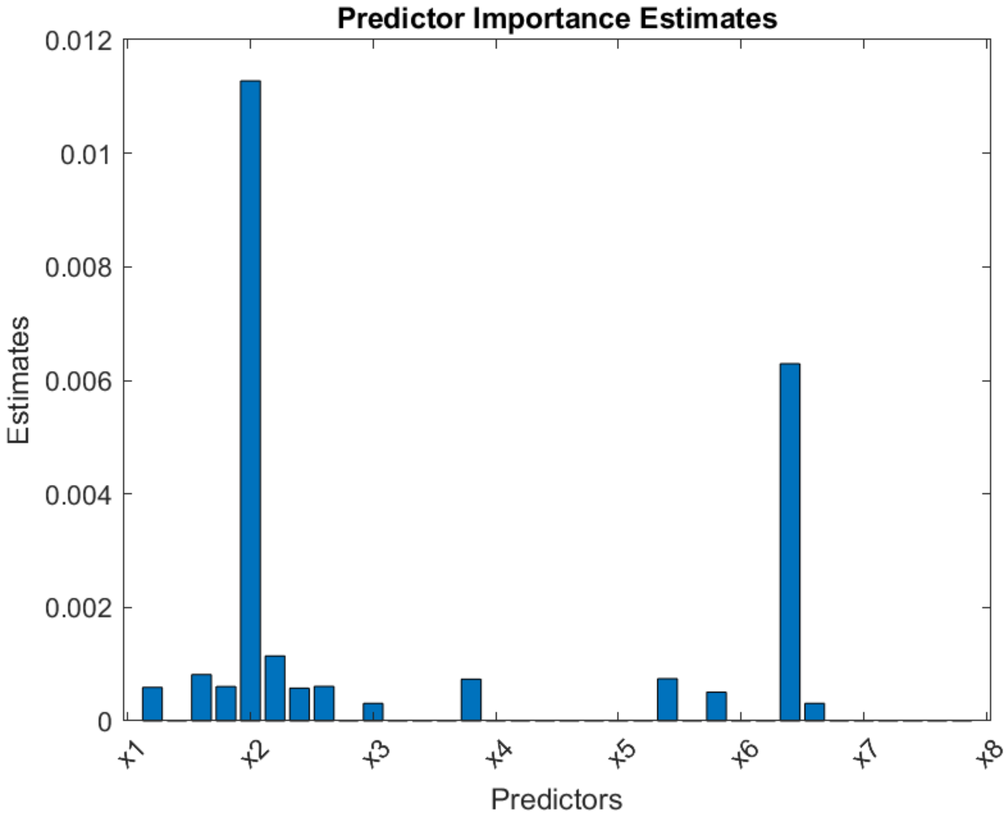
- `tree = fitctree(Tbl,ResponseVarName)` returns a fitted binary classification decision tree based on the input variables (also known as predictors, features, or attributes) contained in the table `Tbl` and output (response or labels) contained in `Tbl.ResponseVarName`. The returned binary tree splits branching nodes based on the values of a column of `Tbl`.
- `tree = fitctree(X,Y)` returns a fitted binary classification decision tree based on the input variables contained in matrix `X` and output `Y`. The returned binary tree splits branching nodes based on the values of a column of `X`.

Classification Tree in MATLAB



Classification Tree in MATLAB

- Estimate predictor importance values by summing changes in the risk due to splits on every predictor and dividing the sum by the number of branch nodes.
- Compare the estimates using a bar graph.
- `imp = predictorImportance(tc);`



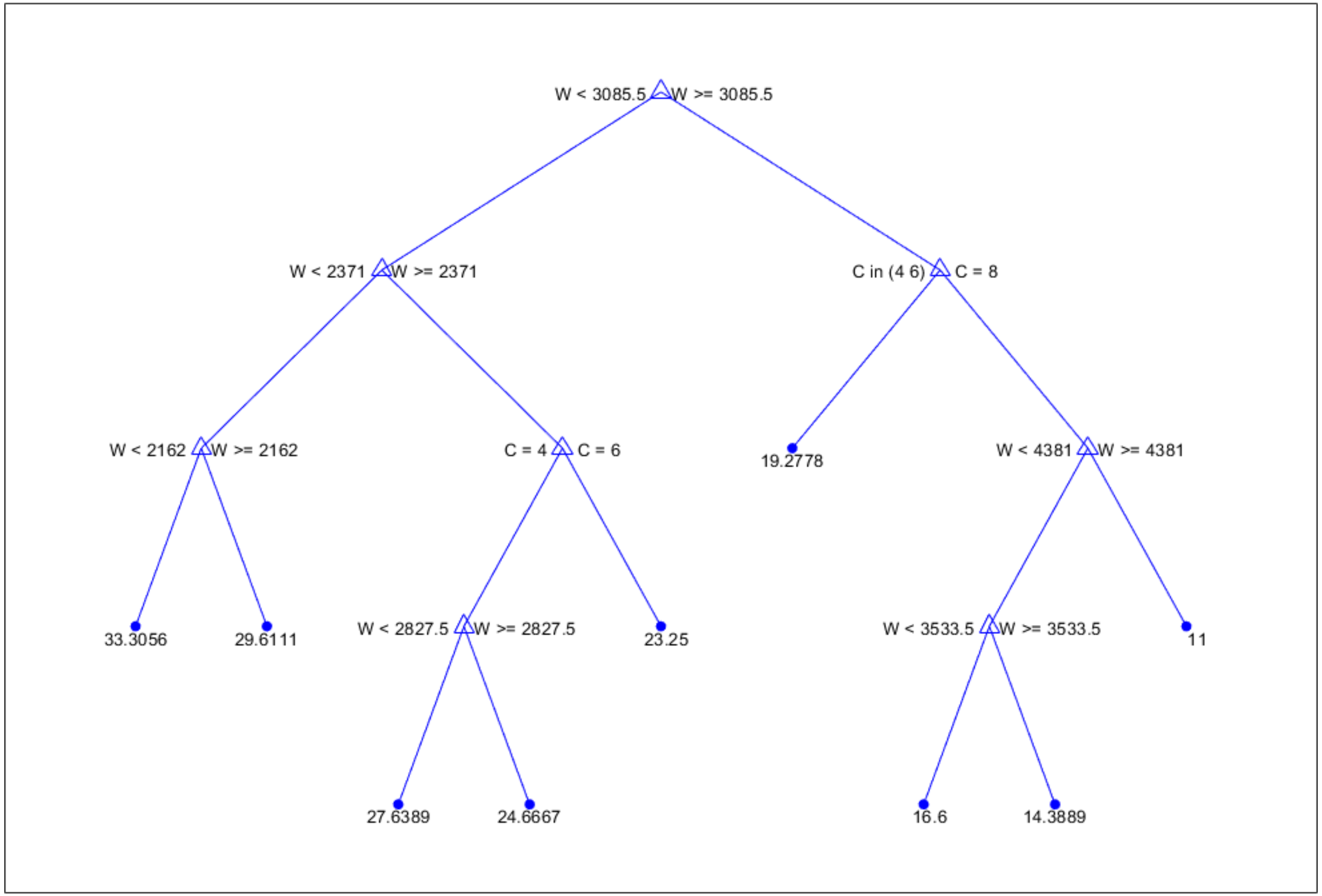
Regression Tree in MATLAB

- **Class:** `RegressionTree`
- Superclasses: `CompactRegressionTree`
- Regression tree
- Description
  - A decision tree with binary splits for regression. An object of class `RegressionTree` can predict responses for new data with the predict method. The object contains the data used for training, so can compute resubstitution predictions.
- Construction
  - Create a `RegressionTree` object by using `fitrtree`.

Regression Tree in MATLAB

- `tree = fitrtree(Tbl,ResponseVarName)` returns a regression tree based on the input variables (also known as predictors, features, or attributes) in the table `Tbl` and the output (response) contained in `Tbl.ResponseVarName`. The returned tree is a binary tree where each branching node is split based on the values of a column of `Tbl`.
- `tree = fitrtree(X,Y)` returns a regression tree based on the input variables `X` and the output `Y`. The returned tree is a binary tree where each branching node is split based on the values of a column of `X`.

Regression Tree in MATLAB



Ensemble Learning

- Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
- Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives.
- Supervised learning algorithms perform the task of searching through a hypothesis space to find a suitable hypothesis that will make good predictions with a particular problem.
- Even if the hypothesis space contains hypotheses that are very well-suited for a particular problem, it may be very difficult to find a good one. Ensembles combine multiple hypotheses to form a (hopefully) better hypothesis.

Random Forests

- Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Random decision forests correct for decision trees' habit of overfitting to their training set.
- The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

Random Forests in MATLAB

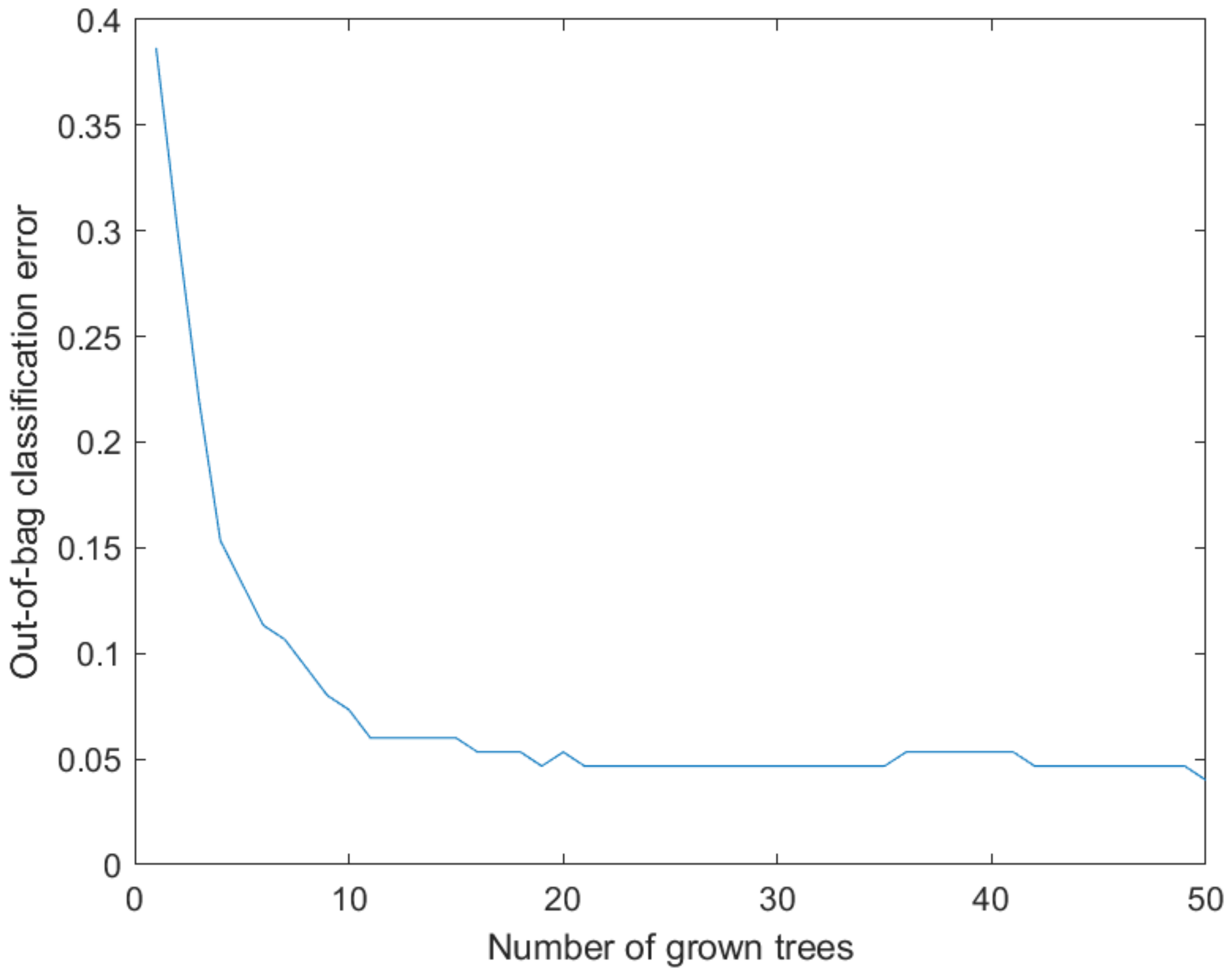
Class: `TreeBagger`

- Create bag of decision trees
- Individual decision trees tend to overfit. Bootstrap-aggregated (bagged) decision trees combine the results of many decision trees, which reduces the effects of overfitting and improves generalization. TreeBagger grows the decision trees in the ensemble using bootstrap samples of the data. Also, TreeBagger selects a random subset of predictors to use at each decision split as in the random forest algorithm.
- By default, TreeBagger bags classification trees. To bag regression trees instead, specify 'Method','regression'.
- For regression problems, TreeBagger supports mean and quantile regression (that is, quantile regression forest).

Random Forests in MATLAB

- `Mdl = TreeBagger(NumTrees,Tbl,ResponseVarName)` returns an ensemble of `NumTrees` bagged classification trees trained using the sample data in the table `Tbl`. ResponseVarName is the name of the response variable in `Tbl`.
- `Mdl = TreeBagger(NumTrees,Tbl,Y)` returns an ensemble of classification trees using the predictor variables in table `Tbl` and class labels in vector `Y`.
- `Y` is an array of response data. Elements of `Y` correspond to the rows of `Tbl`. For classification, `Y` is the set of true class labels. Labels can be any grouping variable, that is, a numeric or logical vector, character matrix, string array, cell array of character vectors, or categorical vector. `TreeBagger` converts labels to a cell array of character vectors. For regression, `Y` is a numeric vector. To grow regression trees, you must specify the name-value pair `'Method','regression'`.
- `B = TreeBagger(NumTrees,X,Y)` creates an ensemble `B` of `NumTrees` decision trees for predicting response `Y` as a function of predictors in the numeric matrix of training data, `X`. Each row in `X` represents an observation and each column represents a predictor or feature.

Random Forests in MATLAB



References

- MATLAB Documentation
  - Decision Trees.
    URL: https://www.mathworks.com/help/stats/decision-trees.html
  - Classification Trees.
    URL: https://www.mathworks.com/help/stats/classificationtree-class.html
  - Regression Trees.
    URL: https://www.mathworks.com/help/stats/regressiontree-class.html
  - Bag of Decision Trees.
    URL: https://www.mathworks.com/help/stats/treebagger.html