

Random Forest

Predict using an ensemble of decision trees.

Inputs

- Data: input dataset
- Preprocessor: preprocessing method(s)

Outputs

- Learner: random forest learning algorithm
- Model: trained model

Random forest is an ensemble learning method used for classification, regression and other tasks. It was first proposed by Tin Kam Ho and further developed by Leo Breiman (Breiman, 2001) and Adele Cutler.

Random Forest builds a set of decision trees. Each tree is developed from a bootstrap sample from the training data. When developing individual trees, an arbitrary subset of attributes is drawn (hence the term “Random”), from which the best attribute for the split is selected. The final model is based on the majority vote from individually developed trees in the forest.

Random Forest works for both classification and regression tasks.

The screenshot shows the 'Random Forest' widget interface in Orange Data Mining. The window has a title bar with three colored buttons (red, yellow, grey) and the text 'Random Forest'. Below the title bar, there is a 'Name' field with the text 'Random Forest' and a circled '1' next to it. Below the 'Name' field, there is a 'Basic Properties' section with a circled '2' next to it. Inside the 'Basic Properties' section, there are two settings: 'Number of trees:' with a value of '10' and 'Number of attributes considered at each split:' with a value of '5'. Both values are in input boxes with up and down arrows.

1. Specify the name of the model. The default name is “Random Forest”.
2. Specify how many decision trees will be included in the forest (*Number of trees in the forest*), and how many attributes will be arbitrarily drawn for consideration at each node. If the latter is not specified (option *Number of attributes...* left unchecked), this number is equal to the square root of the number of attributes in the data. You can also choose to fix the seed for tree generation (*Fixed seed for random generator*), which enables replicability of the results.
3. Original Breiman’s proposal is to grow the trees without any pre-pruning, but since pre-pruning often works quite well and is faster, the user can set the depth to which the trees will be grown (*Limit depth of individual trees*). Another pre-pruning option is to select the smallest subset that can be split (*Do not split subsets smaller than*).
4. Produce a report.
5. Click *Apply* to communicate the changes to other widgets. Alternatively, tick the box on the left side of the *Apply* button and changes will be communicated automatically.

Examples

For classification tasks, we use *iris* dataset. Connect it to **Predictions**. Then, connect **File** to **Random Forest** and **Tree** and connect them further to **Predictions**. Finally, observe the predictions for the two models.

Random Forest

Name: Random Forest

Basic Properties

Number of trees: 10

☐ Number of attributes considered at each split: 5

☐ Fixed seed for random generator: 0

Growth Control

☐ Limit depth of individual trees: 3

☒ Do not split subsets smaller than: 5

Report ☒ Apply Automatically

Predictions

Info

Data: 150 instances.
Predictors: 2
Task: Classification

Restore Original Order

Show

☒ Predicted class
☒ Predicted probabilities for:

Iris-setosa
Iris-versicolor
Iris-virginica

☒ Draw distribution bars

Data View

☒ Show full data set

Output

☒ Original data
☒ Predictions
☒ Probabilities

Report

	Random Forest	Tree	iris	sepal length
71	0.00 : 0.67 : 0.33 → Iris-versi...	0.00 : 0.02 : 0.98 → Iris-virgi...	Iris-versicolor	5.900
72	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.100
73	0.00 : 0.61 : 0.39 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.300
74	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.100
75	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.400
76	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.600
77	0.00 : 0.97 : 0.03 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.800
78	0.00 : 0.71 : 0.29 → Iris-versi...	0.00 : 0.67 : 0.33 → Iris-vers...	Iris-versicolor	6.700
79	0.00 : 0.99 : 0.01 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.000
80	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.700
81	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.500
82	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.500
83	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.800
84	0.00 : 0.33 : 0.67 → Iris-virgi...	0.00 : 0.67 : 0.33 → Iris-vers...	Iris-versicolor	6.000
85	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.400
86	0.00 : 0.89 : 0.11 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.000
87	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.700
88	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.300
89	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.600
90	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.500
91	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	5.500
92	0.00 : 1.00 : 0.00 → Iris-versi...	0.00 : 0.98 : 0.02 → Iris-vers...	Iris-versicolor	6.100

For regressions tasks, we will use *housing* data. Here, we will compare different models, namely **Random Forest**, **Linear Regression** and **Constant**, in the **Test & Score** widget.

The screenshot displays the Orange Data Mining software interface. A workflow is visible on the left, consisting of a 'File' widget connected to three model widgets: 'Random Forest', 'Linear Regression', and 'Constant'. These three models are then connected to a 'Test & Score' widget. The 'Test & Score' widget is open, showing the 'Evaluation Results' table. The 'Random Forest' widget settings are also open, showing various parameters.

Test & Score

Sampling

- ☒ Cross validation
 - Number of folds: 10
 - ☒ Stratified
- ☐ Random sampling
 - Repeat train/test: 10
 - Training set size: 66 %
 - ☒ Stratified
- ☐ Leave one out
- ☐ Test on train data
- ☐ Test on test data

Evaluation Results

Method	MSE	RMSE	MAE	R2
Random Forest	11.372	3.372	2.309	0.865
Linear Regression	23.370	4.834	3.376	0.723
Constant	84.644	9.200	6.662	-0.003

Random Forest

Name: Random Forest

Basic Properties

- Number of trees: 10
- ☐ Number of attributes considered at each split: 5
- ☐ Fixed seed for random generator: 0

Growth Control

- ☐ Limit depth of individual trees: 3
- ☒ Do not split subsets smaller than: 5

Report ☒ Apply Automatically

References

Breiman, L. (2001). Random Forests. In Machine Learning, 45(1), 5-32. Available [here](#).