How to visualize a single Decision Tree from the Random Forest in Scikit-Learn (Python)?

June 29, 2020 by Piotr Płoński **Random forest**

The Random Forest is an esemble of Decision Trees. A single Decision Tree can be easily visualized in several different ways. In this post I will show you, how to visualize a Decision Tree from the Random Forest.

First let's train Random Forest model on Boston data set (it is house price regression task available in `scikit-learn`).

```
# Load packages
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.ensemble import RandomForestRegressor
from sklearn import tree
from dtreeviz.trees import dtreeviz # will be used for tree visual
from matplotlib import pyplot as plt
plt.rcParams.update({'figure.figsize': (12.0, 8.0)})
plt.rcParams.update({'font.size': 14})
```

Load the data and train the Random Forest.

```
boston = load_boston()
X = pd.DataFrame(boston.data, columns=boston.feature_names)
y = boston.target
```

```
rf = RandomForestRegressor(n_estimators=100)
rf.fit(X, y)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='ms
                      max_depth=None, max_features='auto', max_lea
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_lea
                      n_estimators=100, n_jobs=None, oob_score=Fal
                      random_state=None, verbose=0, warm_start=Fal
```

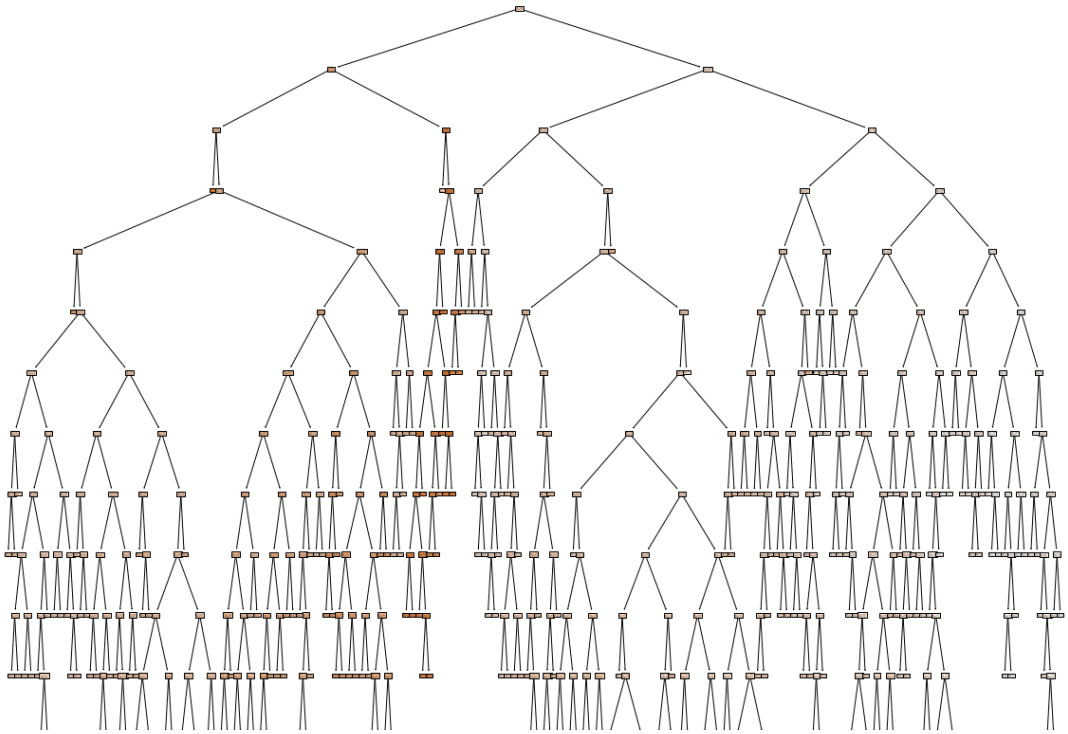
Decision Trees are stored in a `list` in the `estimators_` attribute in the `rf` model. We can check the length of the list, which should be equal to `n_estiamtors` value.

```
len(rf.estimators_)

>>> 100
```

We can plot a first Decision Tree from the Random Forest (with index `0` in the list):

```
plt.figure(figsize=(20,20))
_ = tree.plot_tree(rf.estimators_[0], feature_names=X.columns, fil
```



Do you understand anything? The tree is too large to visualize it in one figure and make it readable.

Let's check the depth of the first tree from the Random Forest:

```
rf.estimators_[0].tree_.max_depth

>>> 16
```

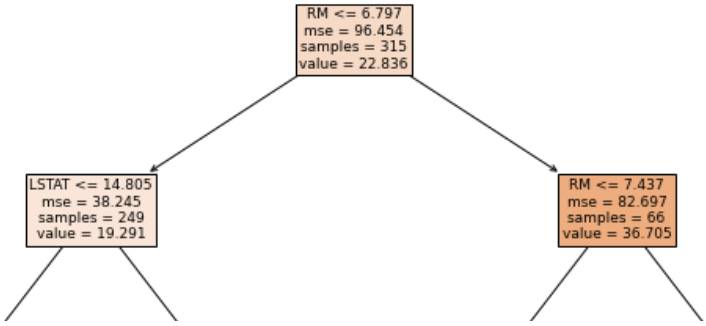
Our first tree has `max_depth=16`. Other trees have similar depth. To make visualization readable it will be good to limit the depth of the tree. In MLJAR's open-source AutoML package mljar-supervised the Decision Tree's depth is set to be in range from 1 to 4. Let's train the Random Forest again with `max_depth=3`.

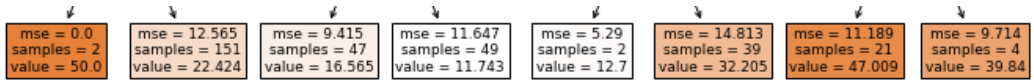
```
rf = RandomForestRegressor(n_estimators=100, max_depth=3)
rf.fit(X, y)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='ms
                      max_depth=3, max_features='auto', max_leaf_n
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_lea
                      n_estimators=100, n_jobs=None, oob_score=Fal
                      random_state=None, verbose=0, warm_start=Fal
```

The plot of first Decision Tree:

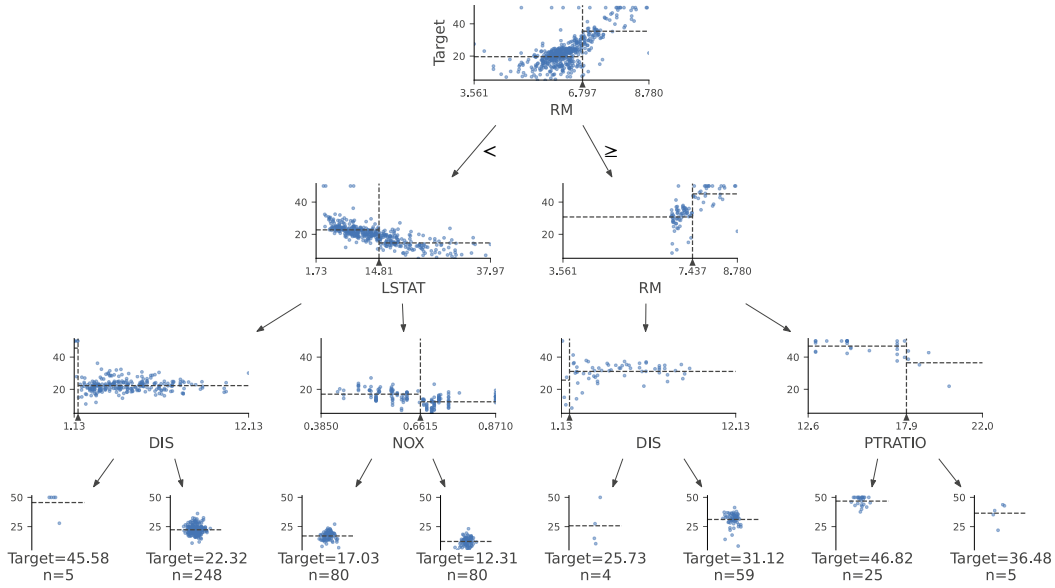
```
_ = tree.plot_tree(rf.estimators_[0], feature_names=X.columns, fil
```





We can use `dtreeviz` package to visualize the first Decision Tree:

```
viz = dtreeviz(rf.estimators_[0], X, y, feature_names=X.columns, t
viz
```



Summary

I show you how to visualize the single Decision Tree from the Random Forest. Trees can be accessed by integer index from `estimators_` list. Sometimes when the tree is too deep, it is worth to limit the depth of the tree with `max_depth` hyper-parameter. What is interesting, limiting the depth of the trees in the Random Forest will make the final model much smaller in terms of used RAM memory and disk space needed to save the model. It will also change the performance of the default Random Forest (with full trees), it will help or not, depending on the data set.

« Random Forest Feature Importance Computed in 3 Ways with Python

How many trees in the Random Forest? »

Articles you might find interesing

1. 8 surprising ways how to use Jupyter Notebook
2. Create a dashboard in Python with Jupyter Notebook
3. Build Computer Vision Web App with Python
4. Develop NLP Web App from Python Notebook
5. Build dashboard in Python with updates and email notifications
6. Share Jupyter Notebook with non-technical users

Join our newsletter

Subscribe to our newsletter to receive product updates

Subscribe

Outstanding Data Science Tools

Blog

About

Brand Assets

GitHub

Twitter

Mercury

AutoML

Pricing

Compare Algorithms

Decision Tree vs Random Forest

Random Forest vs Xgboost

Xgboost vs LightGBM

CatBoost vs Xgboost

AutoML Comparison

What is AutoML?

Golden Features

K-Means Features

Feature Selection