ch02 앙상블 기법- RandomForest(4)

학습 내용

01. 랜덤포레스트 모델의 시각화를 해 보기

```
import sys

if 'google.colab' in sys.modules:
  !pip install -q dtreeviz
```

라이브러리 설치

```
os.getcwd()
```

```
'/content'
```

```
import sys
import os
# add library module to PYTHONPATH
sys.path.append(f"{os.getcwd()}/../")
```

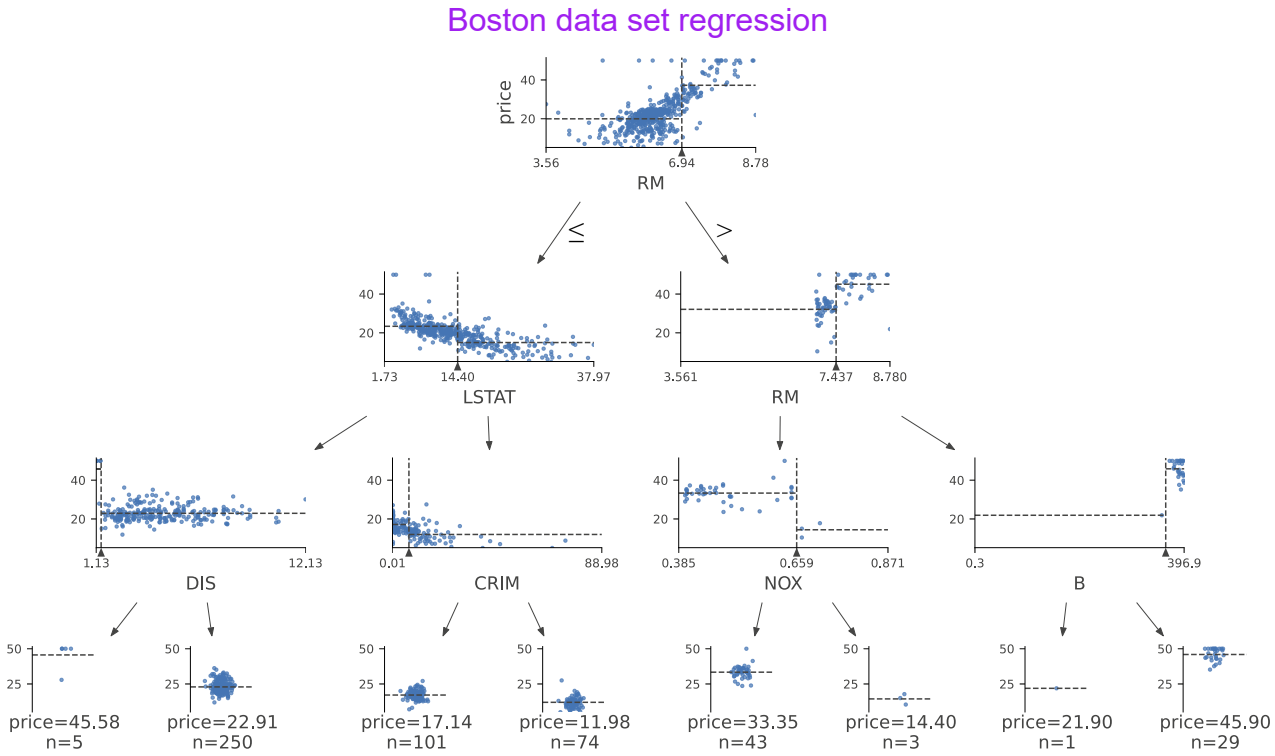
```
from sklearn.datasets import *
from dtreeviz.trees import *
from IPython.display import Image, display_svg, SVG
```

회귀 트리(Regression tree)

- 데이터 셋 : boston data
- url : [boston house-prices dataset](#) (regression).

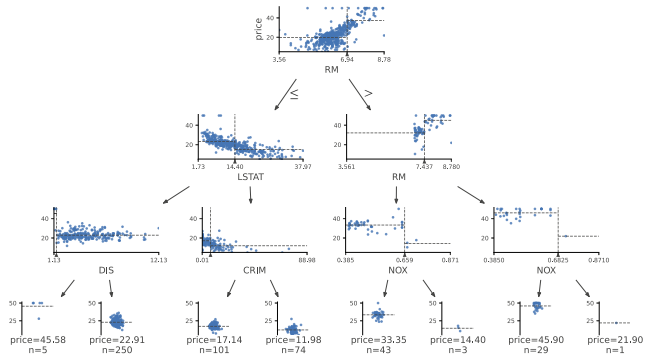
```
model = tree.DecisionTreeRegressor(max_depth=3)
boston = load_boston()

X_train = boston.data
y_train = boston.target
model.fit(X_train, y_train)

viz = dtreeviz(model,
               X_train,
               y_train,
               target_name='price',  # this name will be displayed at the leaf node
               feature_names=boston.feature_names,
               title="Boston data set regression"
```

```
                title= Boston data set regression ",
                fontname="Arial",
                title_fontsize=16,
                colors = {"title":"purple"}
               )
viz
```



이미지 스케일 조정

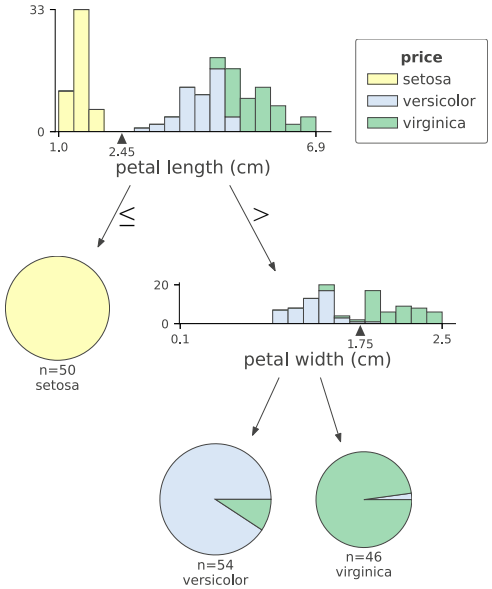
```
dtreeviz(regr,
         X_train,
         y_train,
         target_name='price',  # this name will be displayed at the leaf node
         feature_names=boston.feature_names,
         scale=.5
        )
```



분류 트리(Classification tree)

```
clas = tree.DecisionTreeClassifier(max_depth=2)
iris = load_iris()

X_train = iris.data
y_train = iris.target
clas.fit(X_train, y_train)

viz = dtreeviz(clas,
               X_train,
               y_train,
               target_name='price',
               fontname="Arial",
               feature_names=iris.feature_names,
               class_names=["setosa", "versicolor", "virginica"],
               histtype= 'barstacked')  # barstackes is default
viz
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarning:
  return array(a, dtype, copy=False, order=order)
```



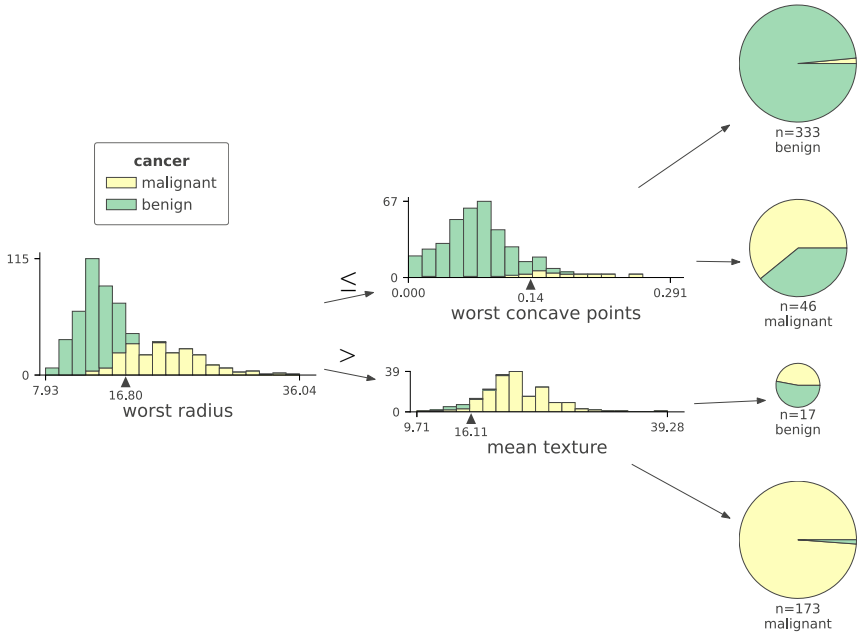
분류 트리(Classification tree)

[Breast Cancer Wisconsin Dataset]

```
model = tree.DecisionTreeClassifier(max_depth=2)
cancer = load_breast_cancer()

X_train = cancer.data
y_train = cancer.target
model.fit(X_train, y_train)

viz = dtreeviz(model,
               X_train,
               y_train,
               target_name='cancer'
```

```
                target_name= 'cancer' ,
                feature_names=cancer.feature_names,
                class_names=["malignant", "benign"],
                orientation='LR')
viz
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarning:
  return array(a, dtype, copy=False, order=order)
```



분류 트리(Classification tree)

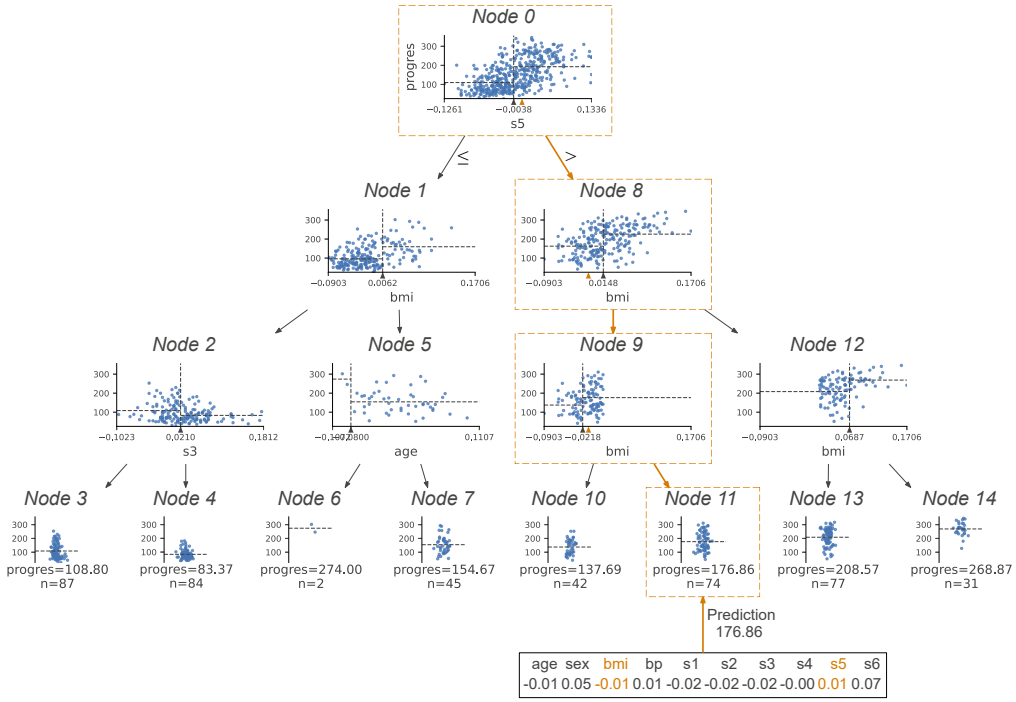
- 데이터 셋 : digits dataset

```
regr = tree.DecisionTreeRegressor(max_depth=3)
diabetes = load_diabetes()
```

컬럼명	설명	데이터 유형
age	나이	숫자
sex	성별	명목형
bmi	체질량 지수	숫자
bp	평균 혈압	숫자
s1	혈청 측정값1	숫자
s2	혈청 측정값2	숫자
s3	혈청 측정값3	숫자
s4	혈청 측정값4	숫자
s5	혈청 측정값5	숫자
s6	혈청 측정값6	숫자
Y	10개변수 측정 후, 당뇨병 진행도	숫자

```
X_train = diabetes.data
y_train = diabetes.target
regr.fit(X_train, y_train)
```

```
regr.fit(X_train, y_train)

X = diabetes.data[np.random.randint(0, len(diabetes.data)),:]

viz = dtreeviz(regr,
               X_train,
               y_train,
               target_name='progres',  # this name will be displayed at the leaf node
               feature_names=diabetes.feature_names,
               X=X,
               show_node_labels = True,
               scale=.7
              )
viz
```




```
import pandas as pd

train = pd.read_csv("house_train.csv")
test = pd.read_csv("house_test.csv")
```

캐글 코리아 2차 대회 데이터 셋 데이터

- https://www.kaggle.com/c/2019-2nd-ml-month-with-kakr/data

컬럼명	의미	값(기타)
ID	집을 구분하는 번호	
date	집을 구매한 날짜	
price	집의 가격(Target variable)	
bedrooms	침실의 수	
bathrooms	화장실의 수	
sqft_living	주거 공간의 평방 피트(면적)	
sqft_lot	부지의 평방 피트(면적)	

컬럼명	의미	값(기타)
floors	집의 층 수	
waterfront	집의 전방에 강이 흐르는지 유무 (a.k.a. 리버뷰)	
view	집이 얼마나 좋아 보이는지의 정도	
condition	집의 전반적인 상태	
grade	King County grading 시스템 기준으로 매긴 집의 등급	
sqft_above	지하실을 제외한 평방 피트(면적)	
sqft_basement	지하실의 평방 피트(면적)	
yr_built	지어진 년도	
yr_renovated	집을 재건축한 년도	
zipcode	우편번호	
lat	위도	
long	경도	
sqft_living15	2015년 기준 주거 공간의 평방 피트(면적, 집을 재건축했다면, 변화가 있을 수 있음)	
sqft_lot15	2015년 기준 부지의 평방 피트(면적, 집을 재건축했다면, 변화가 있을 수 있음)	

```
from sklearn.preprocessing import MinMaxScaler


sel = ['sqft_living', 'sqft_lot', 'bedrooms']  # 'bedrooms' , 'bathrooms',
X = X_all[sel]
y = train['price']

nor_X = MinMaxScaler().fit_transform(X)  # 입력 데이터 정규화
print("정규화 : ", nor_X.shape, y.shape)

#  정규화 데이터 사용
X_train, X_test, y_train, y_test = train_test_split(nor_X, y,
                                                    random_state=42)
```

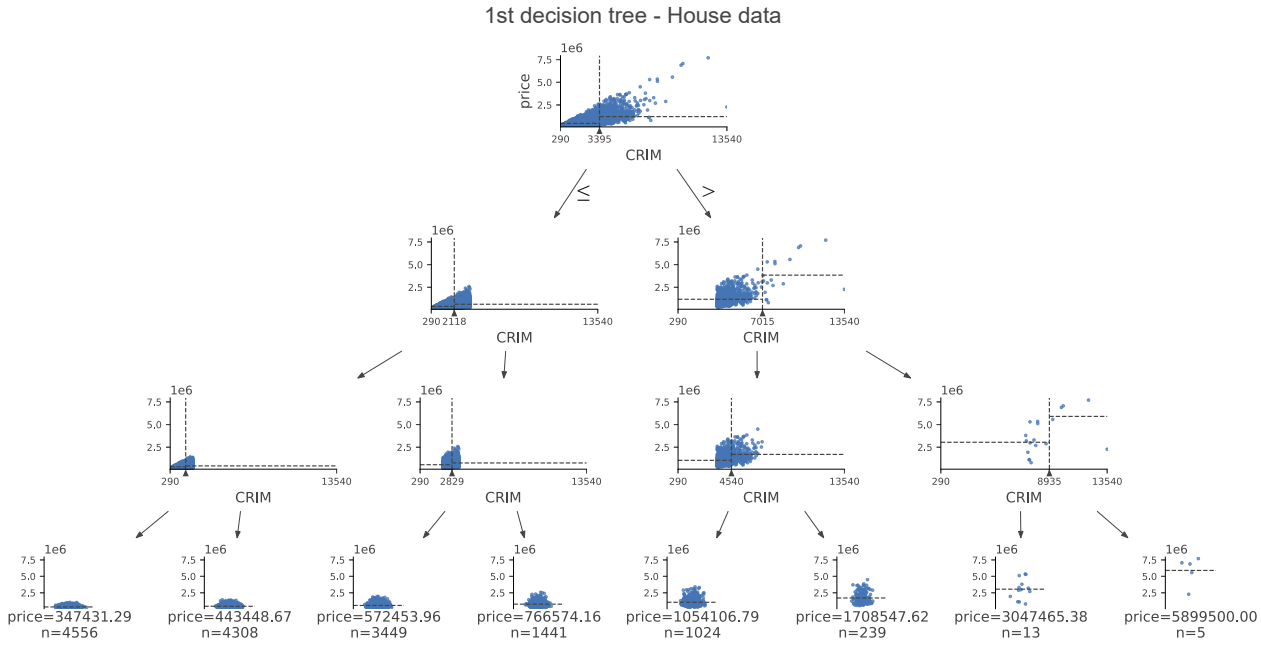
```
정규화 :  (15035, 3) (15035,)
```

```
model = RandomForestRegressor(n_estimators=100,
                              max_depth=3,
                              max_features='auto',
                              min_samples_leaf=4,
                              bootstrap=True,
                              n_jobs=-1,
                              random_state=0)


model.fit(X, y)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=3, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=4,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=-1, oob_score=False,
                      random_state=0, verbose=0, warm_start=False)
```

```
viz = dtreeviz(model.estimators_[0], X, y,
               feature_names=house.feature_names,
               target_name='price',
               fontname="Arial",
               scale=0.8,
               title="1st decision tree - House data")
```

```
viz
```




```
viz.save("decision_tree_house.svg")
```

```
from google.colab import files
files.download("decision_tree_house.svg")
```

REF

- https://colab.research.google.com/github/parrt/dtreeviz/blob/master/notebooks/examples.ipynb
- https://towardsdatascience.com/4-ways-to-visualize-individual-decision-trees-in-a-random-forest-7a9beda1d1b7