

의사결정트리(Decision Tree)

스무고개

- 머신러닝 기법중 지도학습에 해당
- 의사결정 규칙을 나무 구조로 도표화하여 분류와 예측을 수행한다.
- 분류와 회귀에 모두 적용 가능함.
- 장점: 결과가 이해하기 쉽고 직관적이다. 결과에 대한 원인 설명이 필요한 경우에 많이 사용
- 단점: 과적합 가능성이 매우 크기 때문에 정확도가 떨어진다.

온도가 영하입니까?

눈O (15/30)

눈X (15/30)

네

아니오

눈O (13/20)

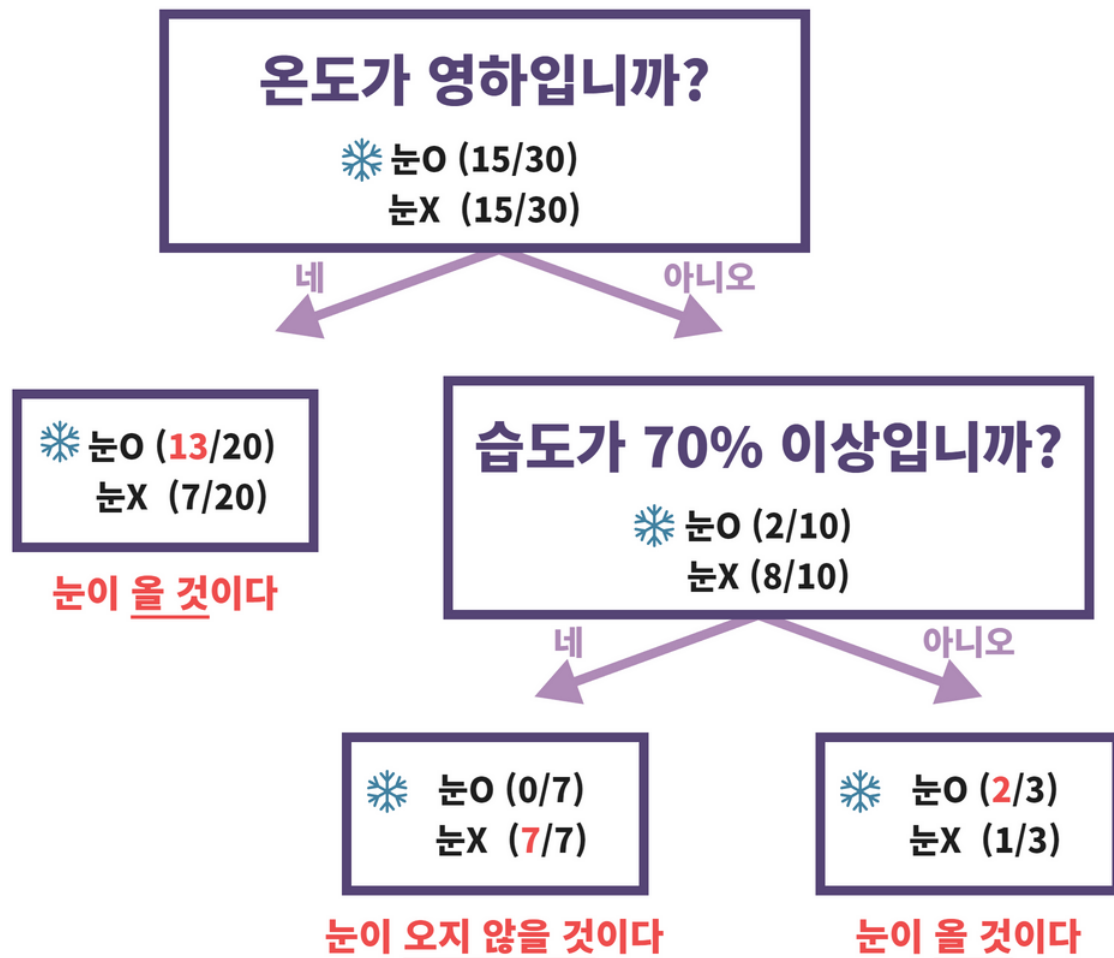
눈X (7/20)

눈이 올 것이다

눈O (2/10)

눈X (8/10)

눈이 오지 않을 것이다



사용 예시

- 은행에서 대출 가능여부 판단
- 환자가 어떠한 조건에 부합하여 병에 걸렸는가에 대한 내용

#01. 패키지 가져오기

`export_graphviz` 함수는 내부적으로 `graphviz` 패키지에 의존하고 있기 때문에 `graphviz` 를 먼저 설치해야 한다.

```
from matplotlib import pyplot as plt
from pandas import read_excel
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_text, export_graphviz, plot_tree
import graphviz
import dtreeviz
from plotnine import *
```

#02. 데이터 가져오기

```
origin = read_excel("https://data.hossam.kr/G02/iris.xlsx")
origin.head()
```

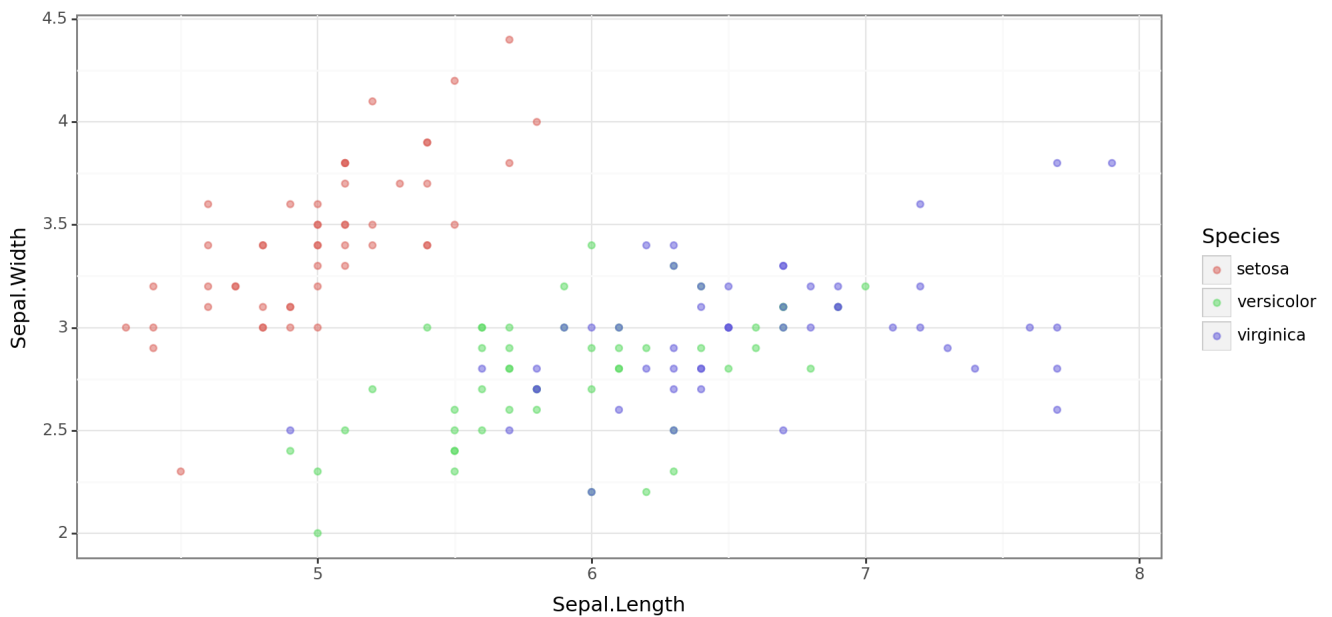
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

#03. 데이터 분할

```
x = origin.drop('Species', axis=1)
y = origin['Species'].map({'setosa':0, 'versicolor':1, 'virginica':2})
x.shape, y.shape
```

```
((150, 4), (150,))
```

```
ggplot(data=origin, mapping=aes(x='Sepal.Length', y='Sepal.Width', color='Species'))
```



<Figure Size: (1000 x 500)>

#04. 학습모델 구현

학습 모델 객체 정의

```
dtree = DecisionTreeClassifier(random_state=777, max_depth=3)
dtree.fit(x.values, y)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=777)
```

결과 확인

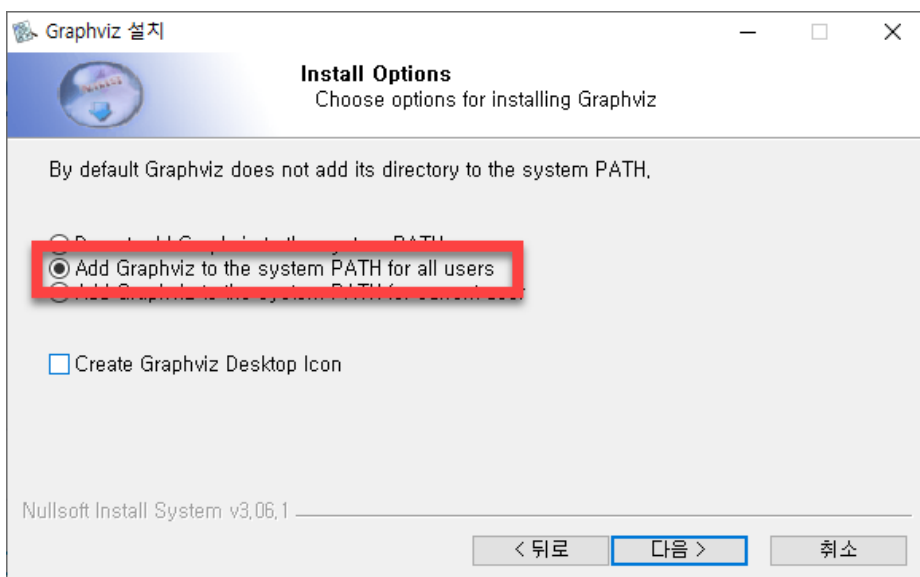
```
et = export_text(dtree,
                 feature_names=list(x.columns),
                 class_names=['setosa', 'versicolor', 'virginica'])
print(et)
```

```
|— Petal.Width ≤ 0.80
|   |— class: setosa
|— Petal.Width > 0.80
|   |— Petal.Width ≤ 1.75
|       |— Petal.Length ≤ 4.95
|           |— class: versicolor
|           |— Petal.Length > 4.95
|               |— class: virginica
|   |— Petal.Width > 1.75
|       |— Petal.Length ≤ 4.85
|           |— class: virginica
|       |— Petal.Length > 4.85
|           |— class: virginica
```

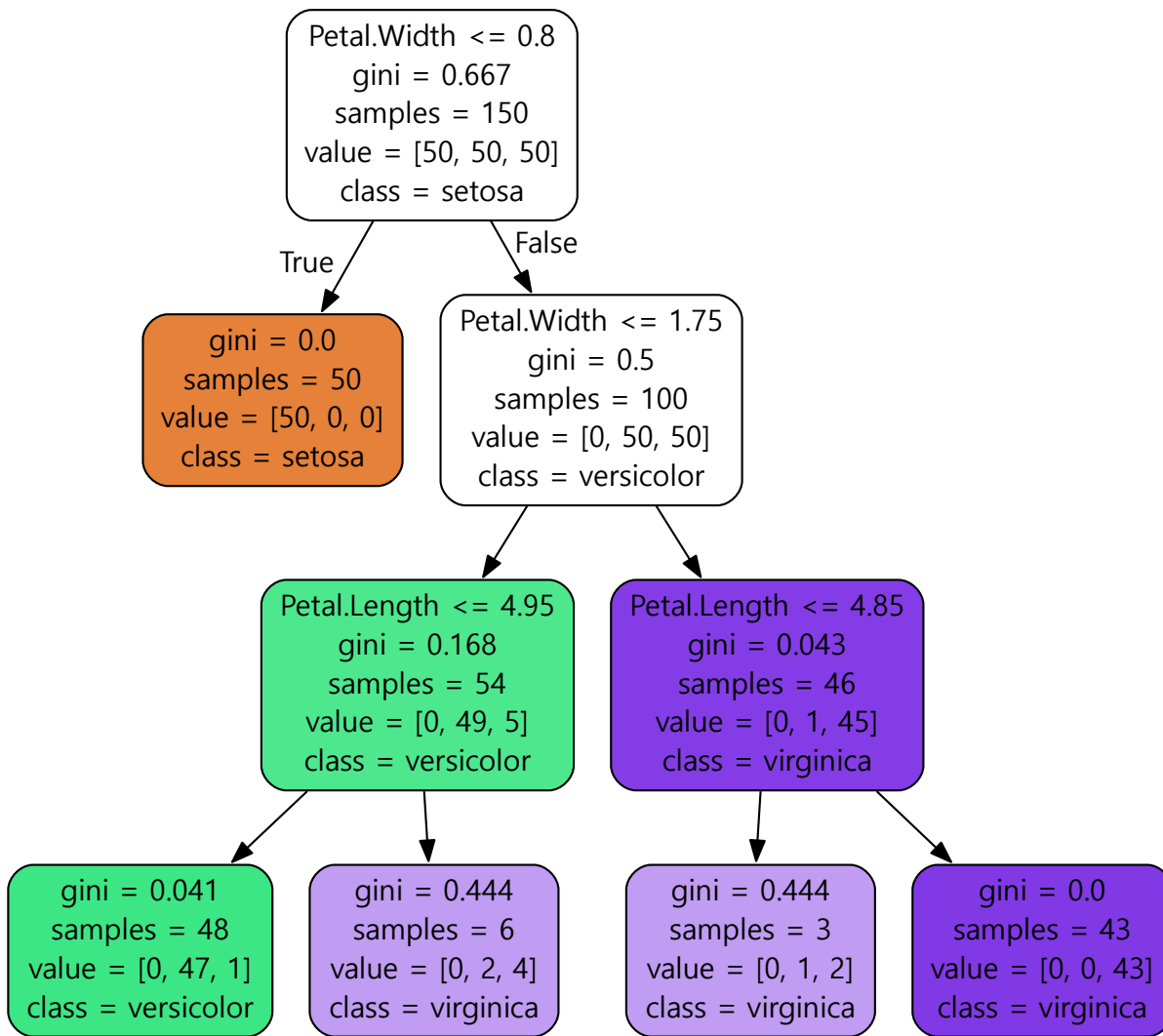
그래픽 시각화

1. Graphviz

1. <https://graphviz.org/download/> 페이지에서 windows 64bit installer 파일을 내려받아 설치한다.
 - 설치과정 중 Install Options 페이지에서 두 번째 항목을 선택하면 PATH 설정이 자동으로 처리된다.



2. 만약 설치 과정중에 PATH 자동 설정을 수행하지 않았다면 설치 후 `C:\\Program Files\\Graphviz\\bin` 경로를 시스템 환경변수 PATH에 추가



2. plot_tree

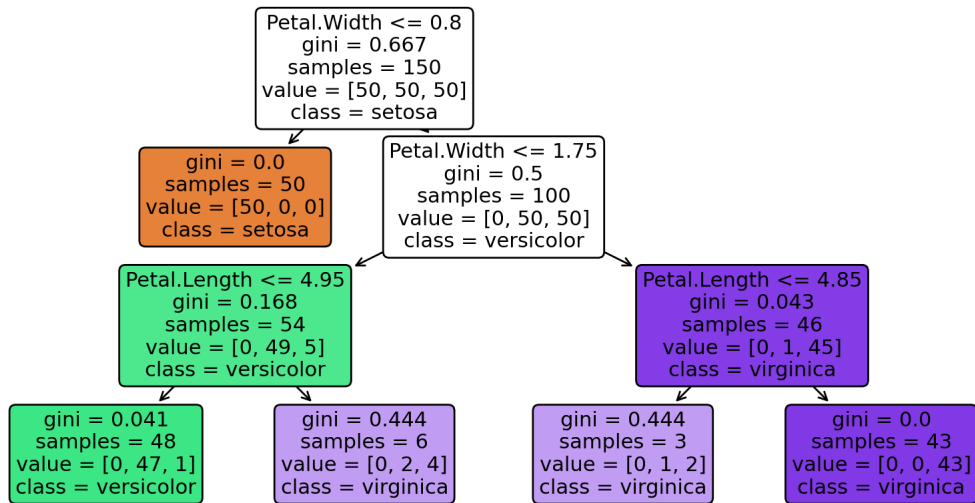
기본 파라미터는 export_text 함수와 동일

하지만 pyplot을 통해서 출력해야 함

```
plt.figure(figsize=(10, 5))

# 리턴을 안받으면 그래프가 2개 출력된다. 리턴값을 따로 사용할 것이 아니므로 언더바로 받는다.
_ = plot_tree(dtree,
              feature_names=list(x.columns),
              class_names=['setosa', 'versicolor', 'virginica'],
              rounded=True,    # 노드의 모서리를 둥글게
              filled=True)    # 노드의 색상을 다르게)

plt.show()
plt.close()
```



3. dtreeviz

`dtreeviz` 패키지의 설치와 참조가 필요하다

```

viz = dtreeviz.model(dtree,
                     X_train=x,
                     y_train=y,
                     target_name="iris",
                     feature_names=list(origin.columns[:-1]),
                     class_names=['setosa', 'versicolor', 'virginica'])

viz.view(scale=2.0)

```