Decision Tree

구름

도시공학과 일반대학원

한양대학교

Decision Tree (의사결정 나무)

CHAID

Kass, G. V. (1980).
"An exploratory technique for investigating large quantities of categorical data". *Applied Statistics*. **29** (2): 119–127.

CART

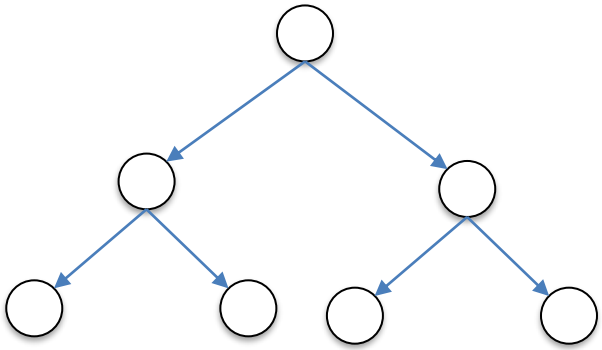
Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984).
Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

ID3

Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106

C4.5

Quinlan, J. R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

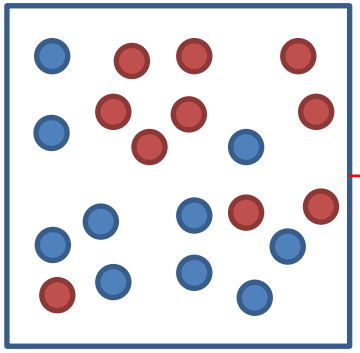


CART : Classification & Regression Tree

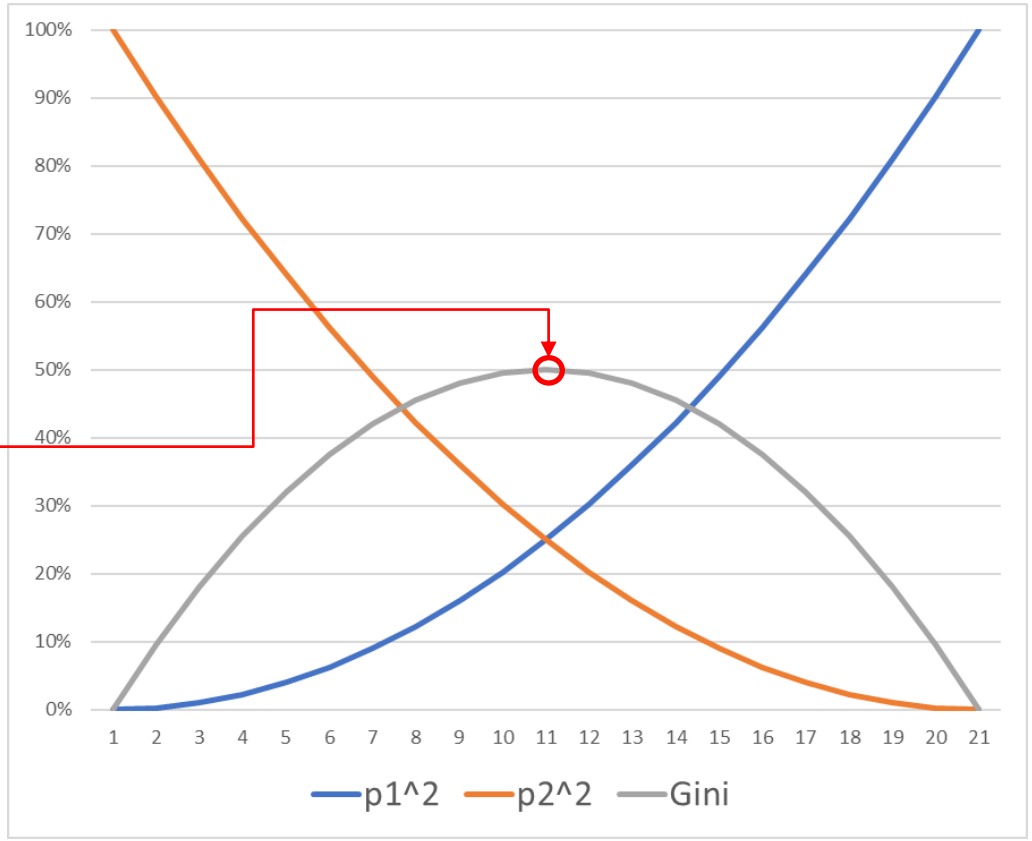
Gini Index(지니인덱스)
Impurity

$$G(S) = 1 - \sum_{i=1}^{c} p_i^2$$

S : 데이터 집합
c : 종속변수 클래스 개수
Pi : 해당 클래스 확률



감소지역 (p1)	비감소지역(p2)
-	20
1	19
2	18
3	17
4	16
5	15
6	14
7	13
8	12
9	11
10	10
11	9
12	8
13	7
14	6
15	5
16	4
17	3
18	2
19	1
20	-

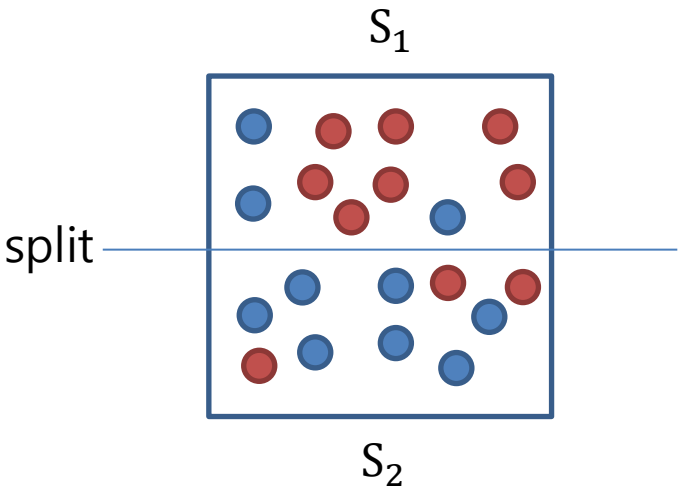


CART : Classification & Regression Tree

Gini Index(지니인덱스)
Impurity

$$I(S) = \sum_{i=0}^{d} R_i \times G(S_i)$$

d : 노드수
Ri : 해당 노드 데이터 비율
Si : 해당 노드 데이터



$$G(S_1) = 1 - \left(\frac{3}{10}\right)^2 - \left(\frac{7}{10}\right)^2 = 0.42$$

$$G(S_2) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2 = 0.42$$

$$I(S) = 0.5 \times 0.42 + 0.5 \times 0.42 = 0.42$$

Information Gain = 0.5 - 0.42 = 0.08

도시계획 현황 통계

http://www.eum.go.kr/web/cp/st/stUpisStatDet.jsp

로그인 | 회원가입

토지e음 토지이용계획 도시계획 규제안내서 고시정보 정보마당 처음오셨나요? 글자크기 + −

도시계획통계

홈 > 도시계획 > 도시계획통계

도시계획현황 통계 | 도시계획현황 분석(KOSIS) | 국토의 계획 및 이용에 관한 연차보고서

개요

「국토의 계획 및 이용에 관한 법률」에 의한 모든 국토의 용도지역·지구·구역, 도시계획시설등에 대하여「도시계획현황」통계책자를 매년 발간하여, 도시행정, 도시정책, 도시연구 및 개발 도시계획 분야에 필요한 기초자료로 제공함

관련내용

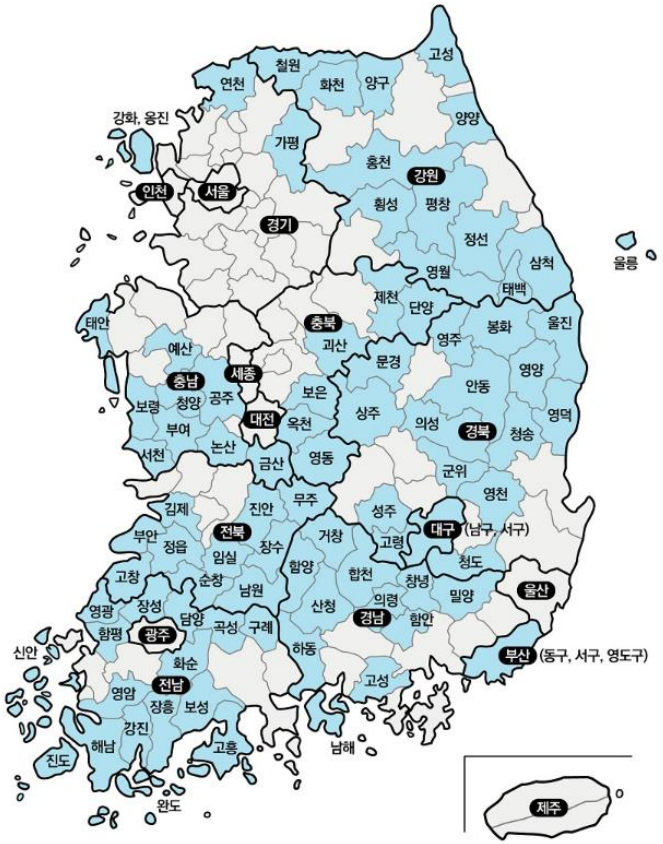
- 화살표 이미지 통계명:도시계획현황통계 | 종류:국가승인통계 | 승인번호:315002 | 작성주기: 1년 | 작성기간:1988년 ~ 현재(2005년까지는 자체생산통계, 2006년 이후부터는 승인통계)
- 관련 사이트:국토교통 통계누리, KOSIS 국가통계포털, e-나라지표

번호	연도	자료명	통계자료	책자	보도자료
1	2022년	2022년 도시계획현황통계	엑셀	PDF	PDF
2	2021년	2021년 도시계획현황통계	엑셀	PDF	PDF
3	2020년	2020년 도시계획현황통계	엑셀	PDF	PDF
4	2019년	2019년 도시계획현황통계	엑셀	PDF	PDF

인구 감소 지역 지정

https://www.mois.go.kr/frt/sub/a06/b06/populationDecline/screen.do

인구감소지역 지정 결과(89개)

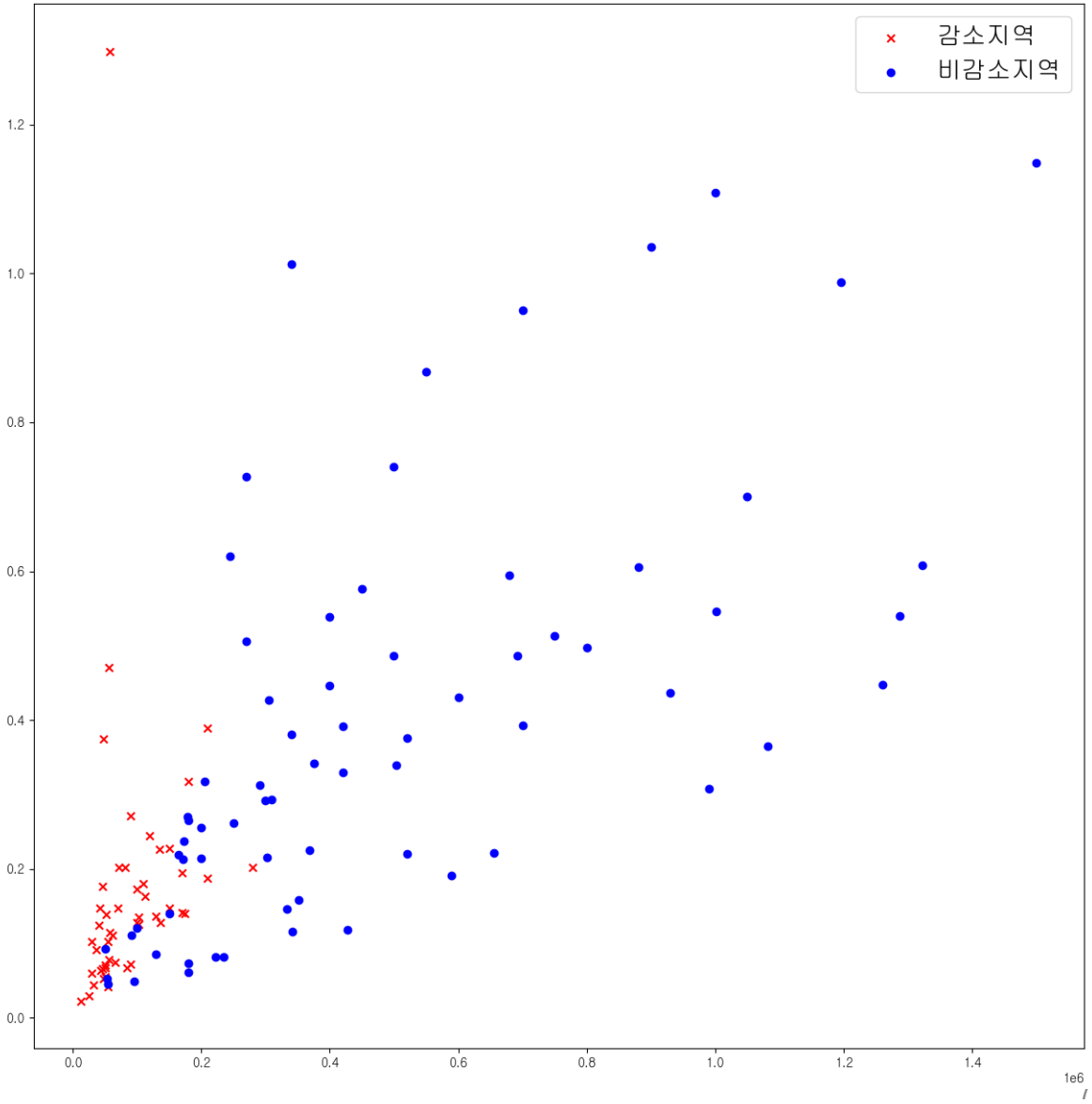


부산(3)	동구 서구 영도구
대구(2)	남구 서구
인천(2)	강화군 옹진군
경기(2)	가평군 연천군
강원(12)	고성군 삼척시 양구군 양양군 영월군 정선군 철원군 태백시 평창군 홍천군 화천군 횡성군
충북(6)	괴산군 단양군 보은군 영동군 옥천군 제천시
충남(9)	공주시 금산군 논산시 보령시 부여군 서천군 예산군 청양군 태안군
전북(10)	고창군 김제시 남원시 무주군 부안군 순창군 임실군 장수군 정읍시 진안군
전남(16)	강진군 고흥군 곡성군 구례군 담양군 보성군 신안군 영광군 영암군 완도군 장성군 장흥군 진도군 함평군 해남군 화순군
경북(16)	고령군 군위군 문경시 봉화군 상주시 성주군 안동시 영덕군 영양군 영주시 영천시 울릉군 울진군 의성군 청도군 청송군
경남(11)	거창군 고성군 남해군 밀양시 산청군 의령군 창녕군 하동군 함안군 함양군 합천군

Sdot Data

Sdot 반경 200m 이내의 지목 면적 비율, 도로(x축)와 대지(y축)

계획인구	시가화용지	감소지역
13000	2291000	1
25000	2946000	1
30000	5947000	1
30000	10215000	1
32000	4392000	1
36500	9117000	1
40500	12423000	1
42000	14760000	1
44000	6317000	1
46000	17707000	1
46000	6560000	1
47000	37500000	1
47000	5234000	1
50000	9252000	0
50000	7062000	1
50000	5565000	1
50000	6729000	1
52500	13901000	1
53600	5224000	0



Gini Index 계산

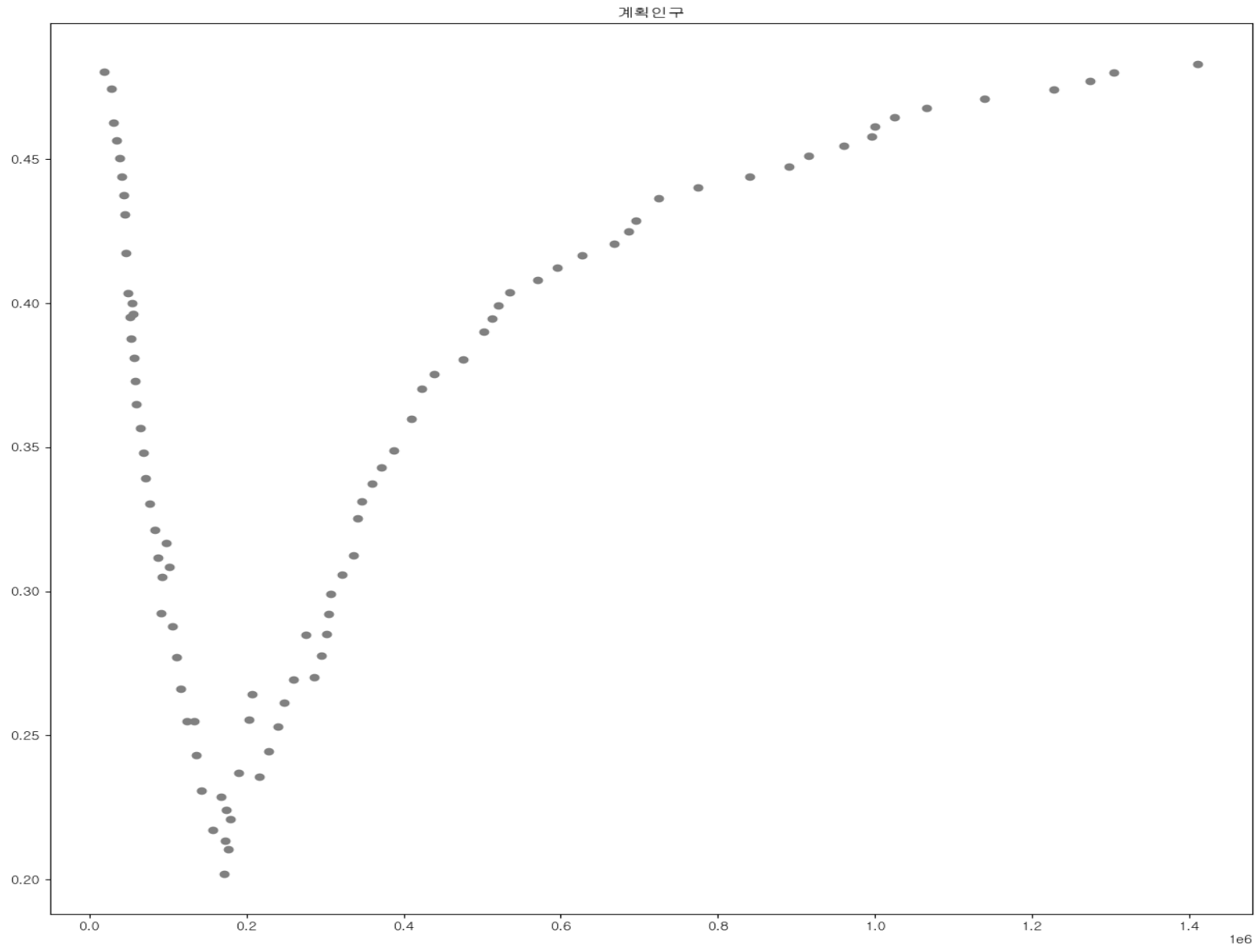
```
"""학습을 위하 pandas를 numpy로 변환하여 x와 y 배열 생성"""
x = np.array(tmp[['계획인구', '시가화용지']].values)
y = np.array(tmp['감소지역'].values)
y = y.reshape(y.shape[0], 1) #x배열과 shape를 같게 reshape

"""지니인덱스 계산 함수"""
7 usages   Kloud_github
def GiniIndex(y):
    total = len(y)
    G = 1
    for c in np.unique(y): #종속변수의 갯수로 loop
        # print(str(c) + "값 : " + str(np.power(np.where(y == c, 1, 0).sum() / total, 2)))
        G = G - np.power(np.where(y == c, 1, 0).sum() / total, 2)
    return G


print(str(GiniIndex(y)))
```

$G(S) =$ 0.4837581124932395

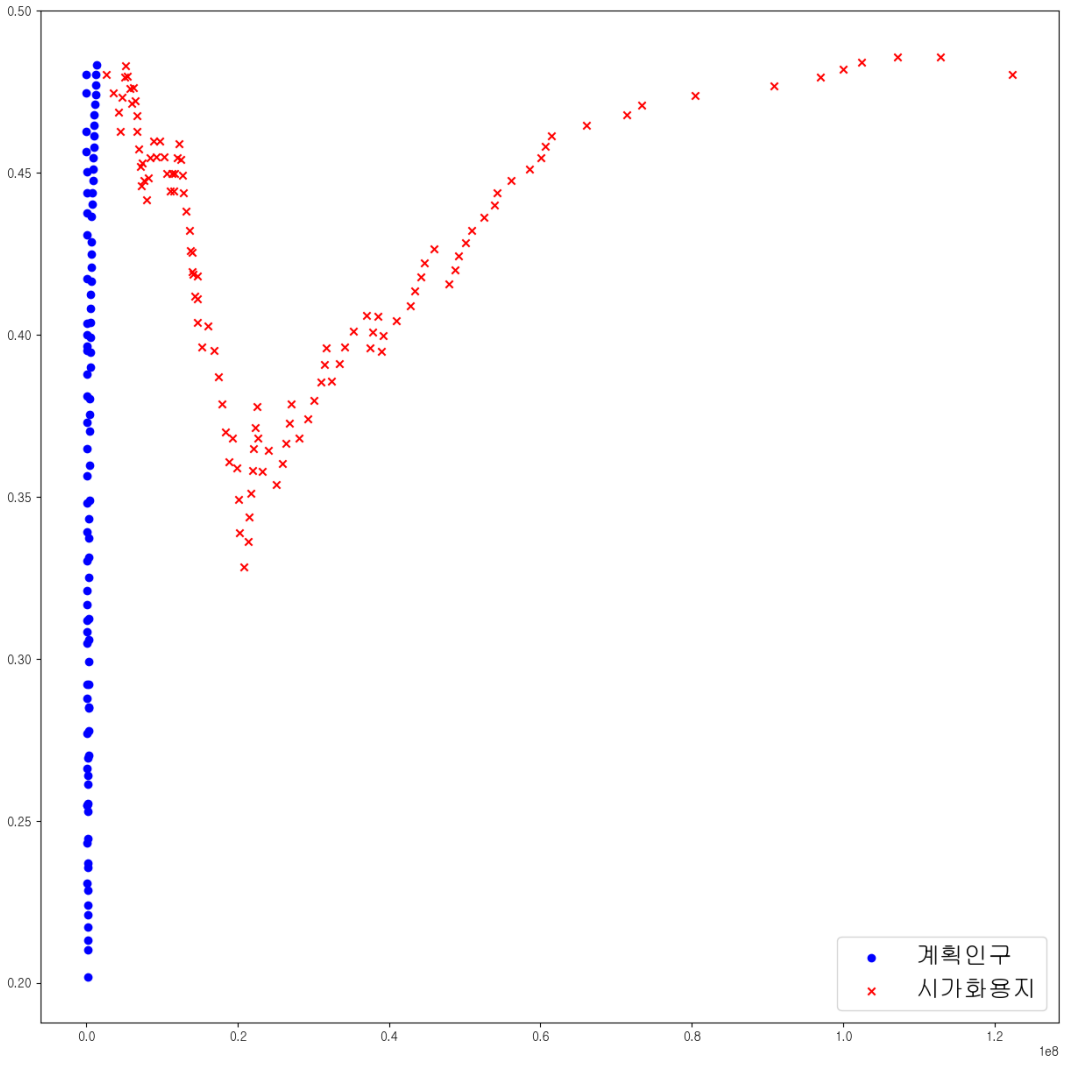
Split Loop

```
criteria = x[:,0]
criteria = np.sort(np.unique(criteria))
total = len(y)
I = np.array([])
for f,l in zip(criteria[:-1], criteria[1:]):
    split = np.mean([f, l])

    s1 = y[np.where(x[:,0] < split, True, False)]
    s2 = y[np.where(x[:,0] > split, True, False)]

    Gini = len(s1) / total *  GiniIndex(s1) + len(s2) / total * GiniIndex(s2)

    I = np.append(I, np.array([f, l, split, Gini]))

I = I.reshape(int(I.shape[0]/4), 4)
```

계획인구에 따른 Gini Index 변화



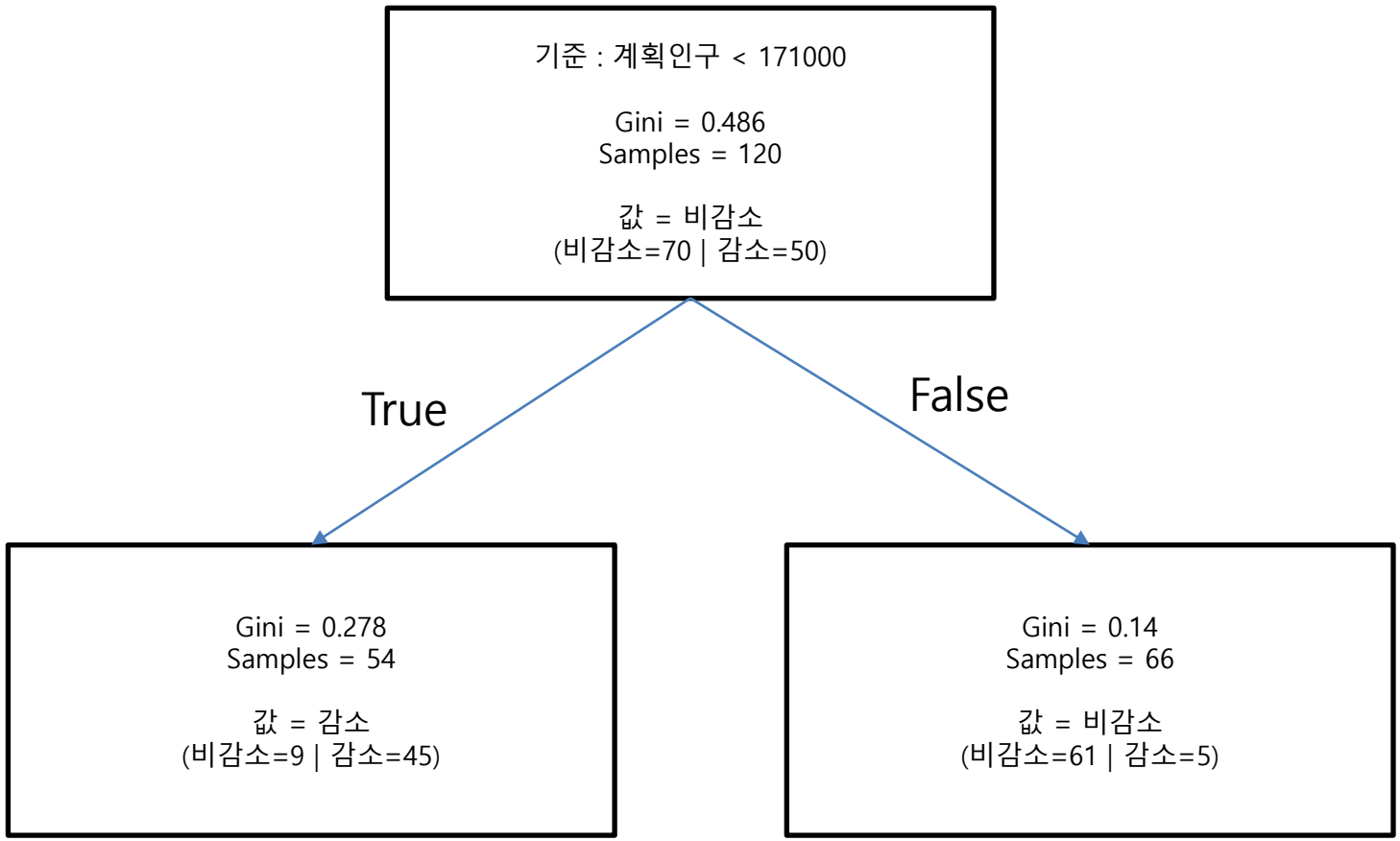
계획인구와 시가화용지에 따른 Gini Index 변화



계획인구 분할시 최저 Gini : 0.2020202020202020

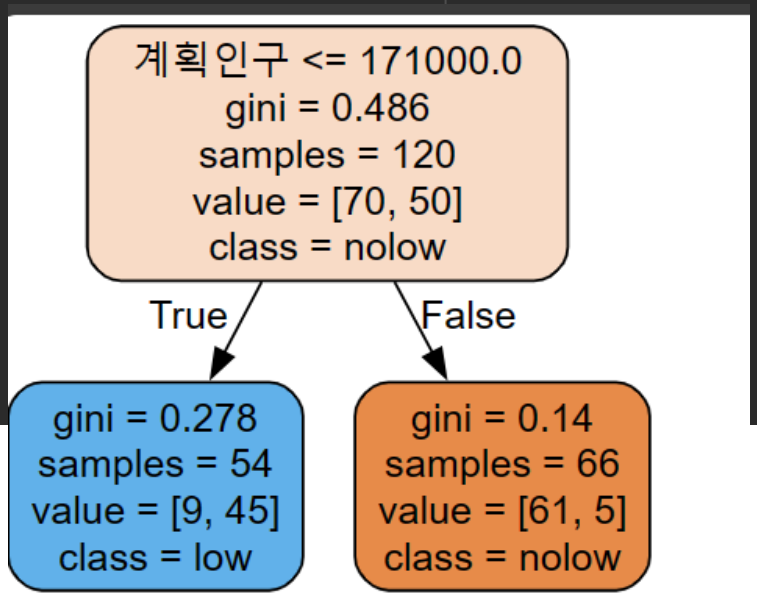
시가화용지 분할시 최저 Gini : 0.32851316277345205

Gini Index Dtree

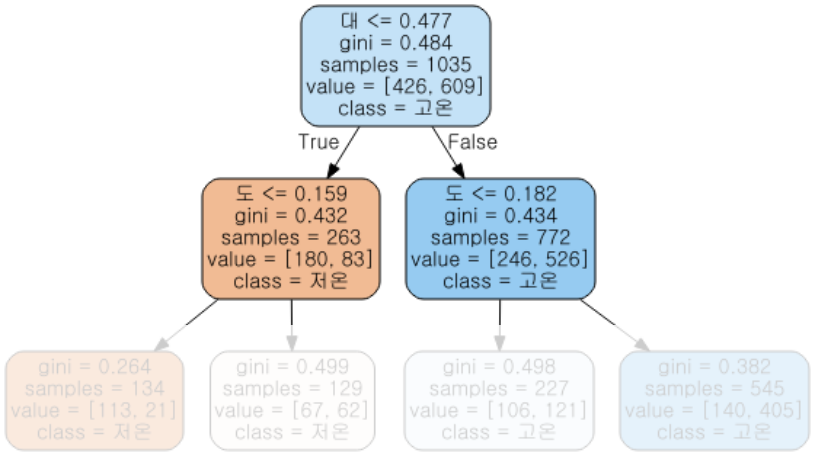


Sklearn 라이브러리 활용시

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
from graphviz import Source
from sklearn.metrics import classification_report, confusion_matrix

tree_clf = DecisionTreeClassifier(max_depth=2    )
tree_clf.fit(x,y)

score_tr = tree_clf.score(x, y)



dt_dot_data   = export_graphviz(tree_clf,
                                feature_names=['계획인구', '시가화용지'],
                                class_names=['nolow', 'low'],          # 종속변수
                                rounded = True,
                                filled = True)



gp = Source(dt_dot_data)

gp.format = 'svg'
img = gp.render('dtree_render',view=True)
```



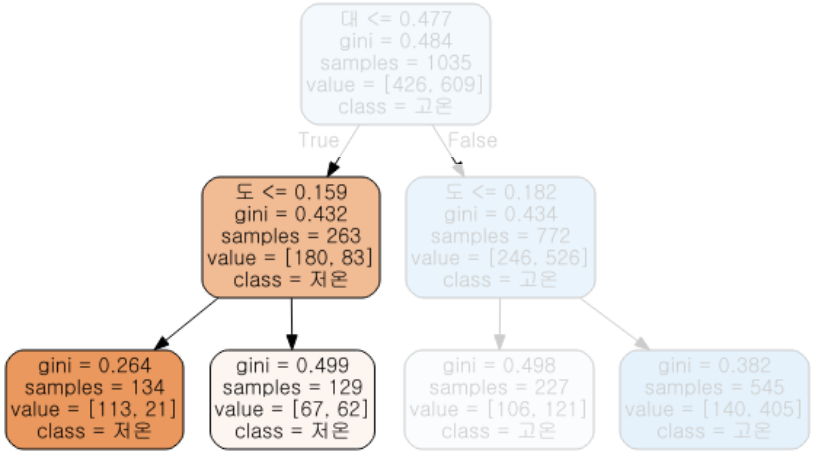
Feature Importance 변수 중요도



Gini = 0.484369

Gini = 0.431985*25.41% + 0.434226*74.59% = 0.433657

Gain = 0.050712 = 대의 중요도

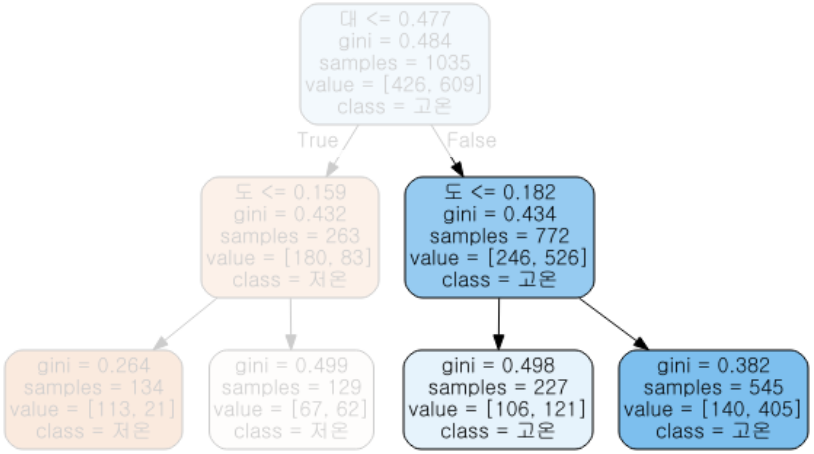


Gini = 0.431985

Gini = 0.264313*50.95% + 0.499249*49.05% = 0.379548

Gain = 0.052437 * 25.41%(parent node) = 0.013325
= 도의 중요도

Feature Importance 변수 중요도



Gini = 0.434226

Gini = 0.497817*29.40% + 0.381786*70.56% = 0.415904

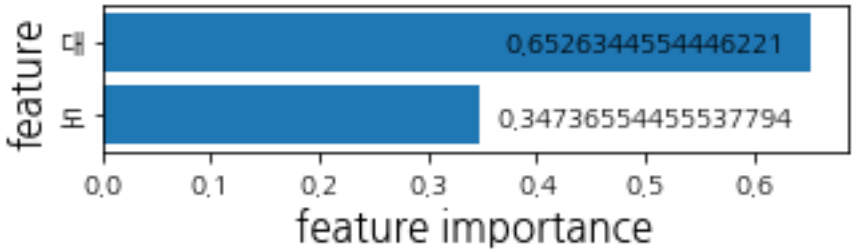
Gain = 0.018322 * 74.59%(parent node) = 0.013666

= 도의 중요도

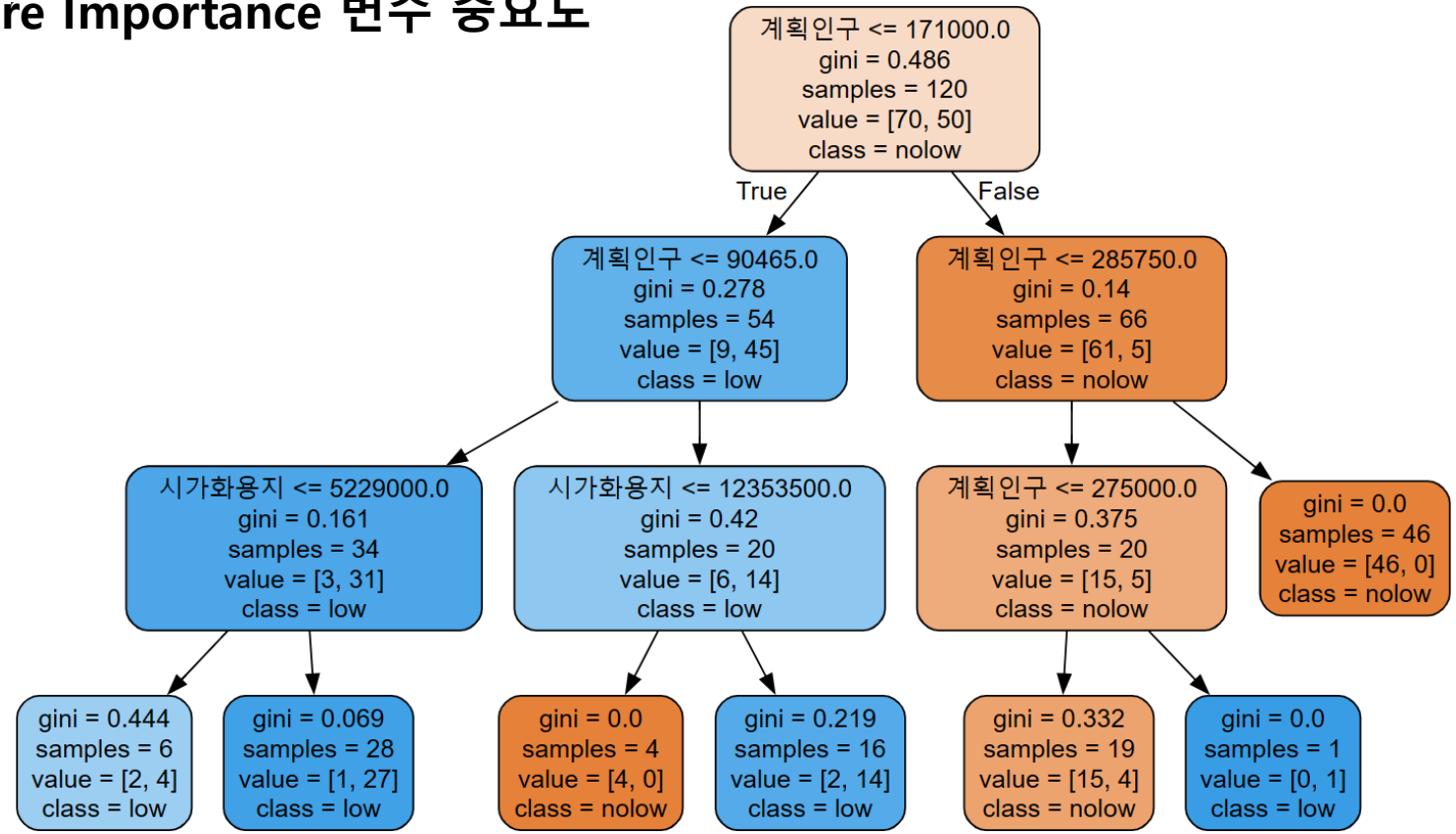
대의 중요도 = 0.050712 = 0.050712 = 65.26%

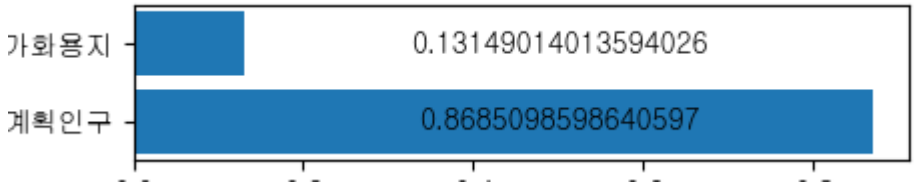
도의 중요도 = 0.013325 + 0.013666 = 0.026991 = 34.73%

0.077703



Feature Importance 변수 중요도



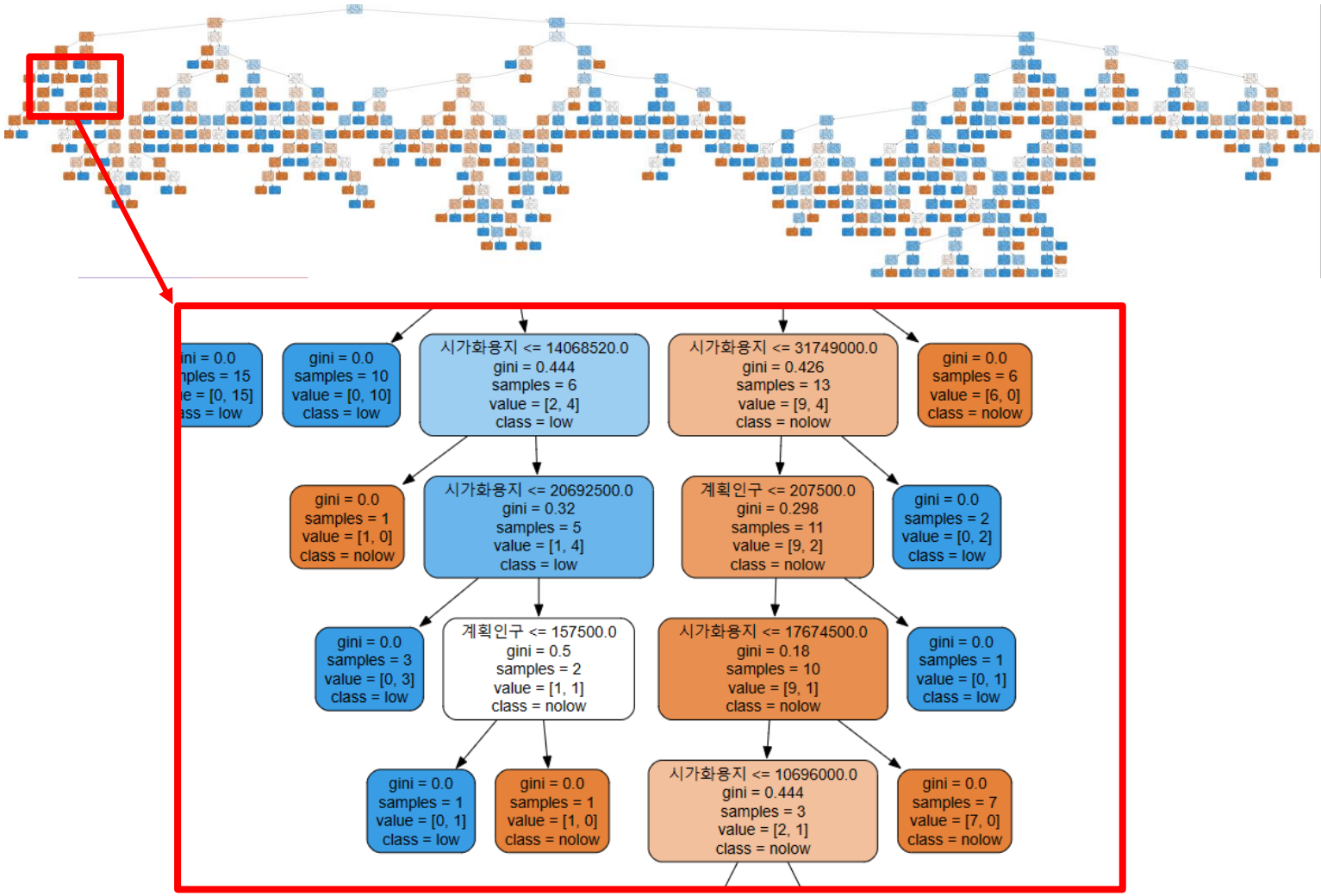


Dtree 과적합 문제를 해결하기위해 = Pruning(가지치기)

Depth = 20

Depth P

0	1
1.0	0.883333...
2.0	0.883333...
3.0	0.925
4.0	0.941666...
5.0	0.975
6.0	0.983333...
7.0	0.991666...
8.0	1.0
9.0	1.0



2진 분류 성능평가지표

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

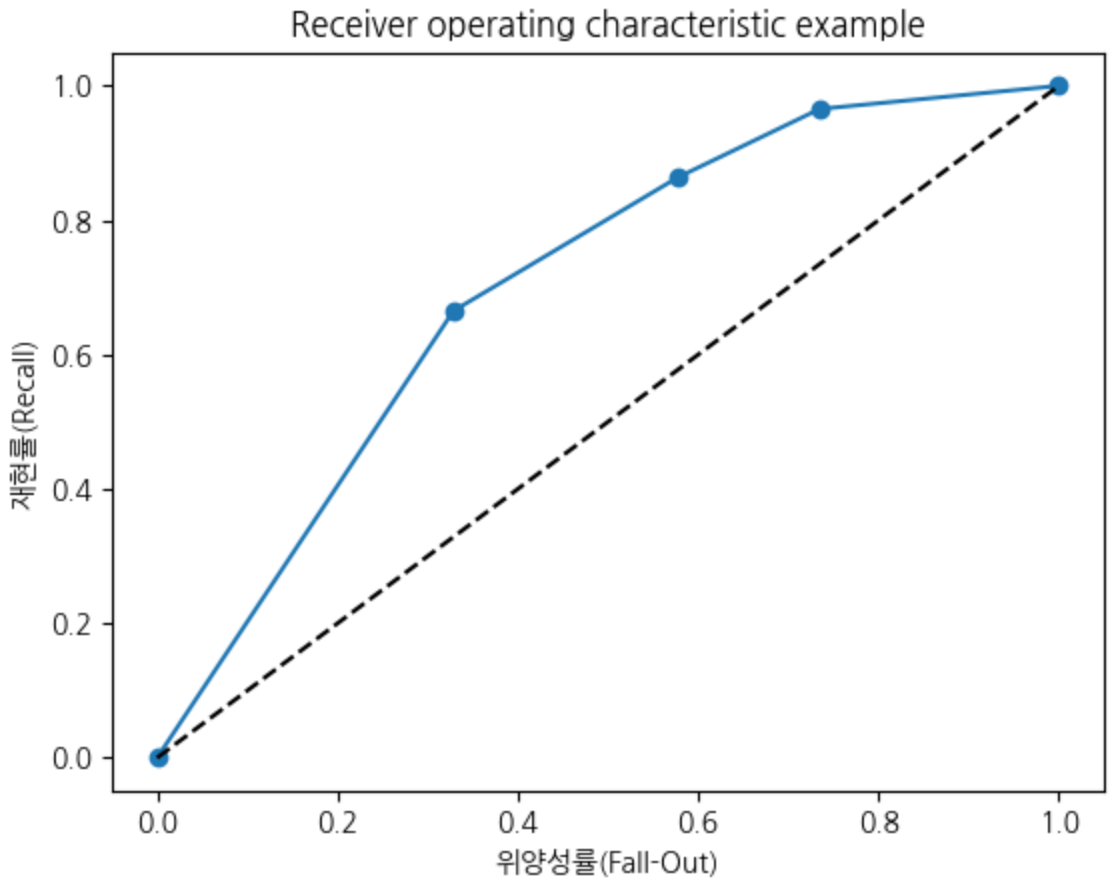
$$(Precision) = \frac{TP}{TP + FP}$$

$$(Recall) = \frac{TP}{TP + FN}$$

$$(Accuracy) = \frac{TP + TN}{TP + FN + FP + TN}$$

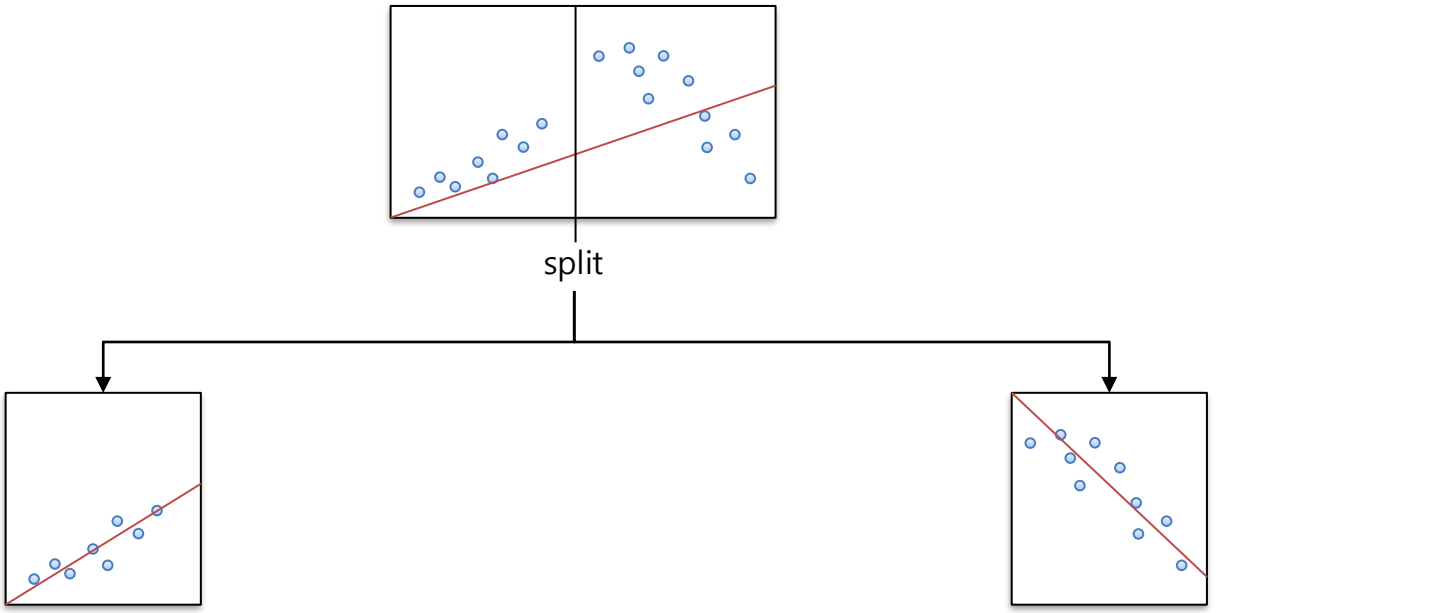
$$(F1\text{-}score) = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

ROC curve / AUC



Regression Tree

$$SSE = \sum_{i=1}^{n}\left((y_i - \hat{y})^2\right) = 100$$



$$SSE_1 = \sum_{i=1}^{n_1}\left((y_i - \hat{y})^2\right) = 12$$

$$SSE_2 = \sum_{i=1}^{n_2}\left((y_i - \hat{y})^2\right) = 13$$

$$GAIN = SSE - (SSE_1 + SSE_2)$$

ID3 entropy (C4.5, C5.0)

$$Entropy(S) = \sum_{i=1}^{c} p_i * I(x_i) \quad , \quad I(x) = log_2 \frac{1}{p(x)}$$