텍 스 트 세 미 나

ToBig's 10기 이준걸

Ensemble 1

Decision Tree

Contents

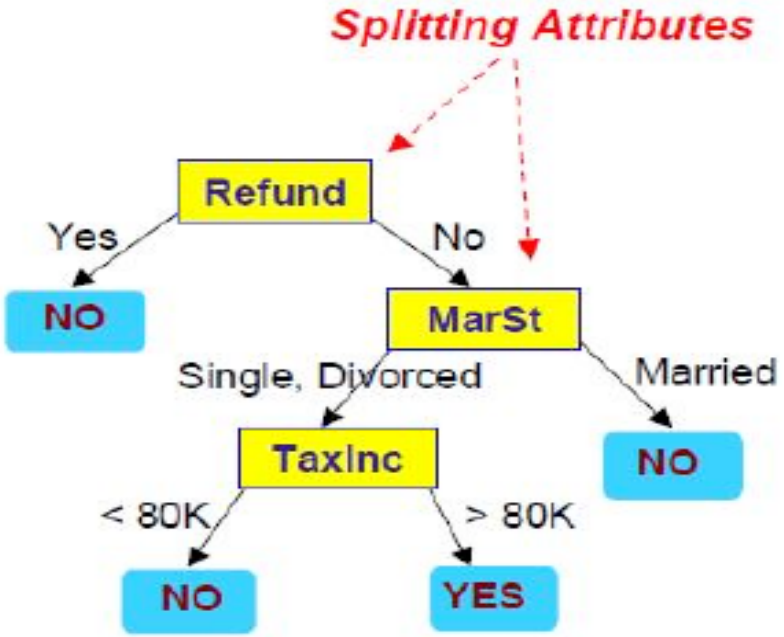
Machine Learning Type

1. Gradient Descent Based Learning
2. Probability Theory Based Learning
3. Information Theory Based Learning
4. Distance Similarity Based Learning

Decision Tree Classifier

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

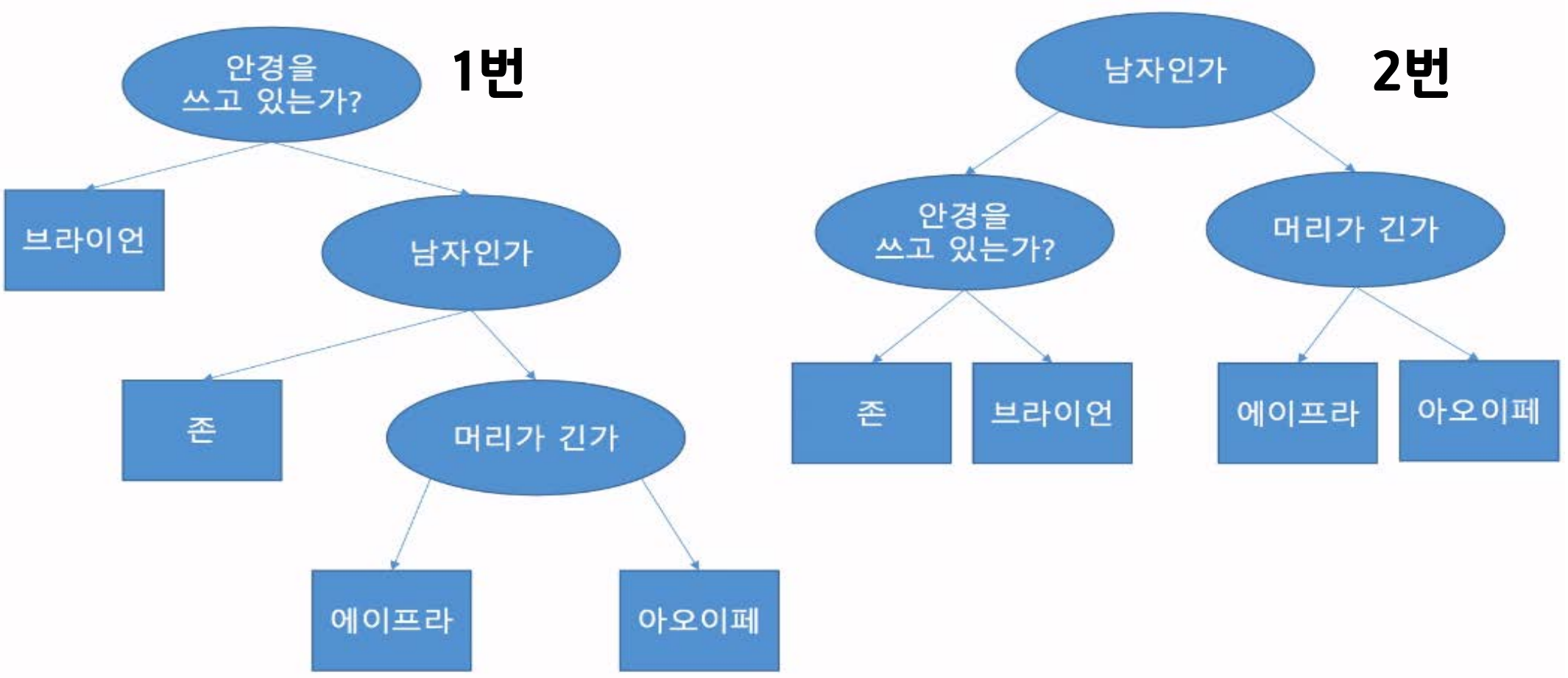
Data를 가장 잘 구분할 수 있는 Tree 만들기

Decision Tree Classifier

남자	긴 머리	안경	이름
예	아니오	예	브라이언
예	아니오	아니오	존
아니오	예	아니오	에이프라
아니오	아니오	아니오	아이오페

Unit 01 | Decision Tree OverView

어떤 트리가 좋을까?



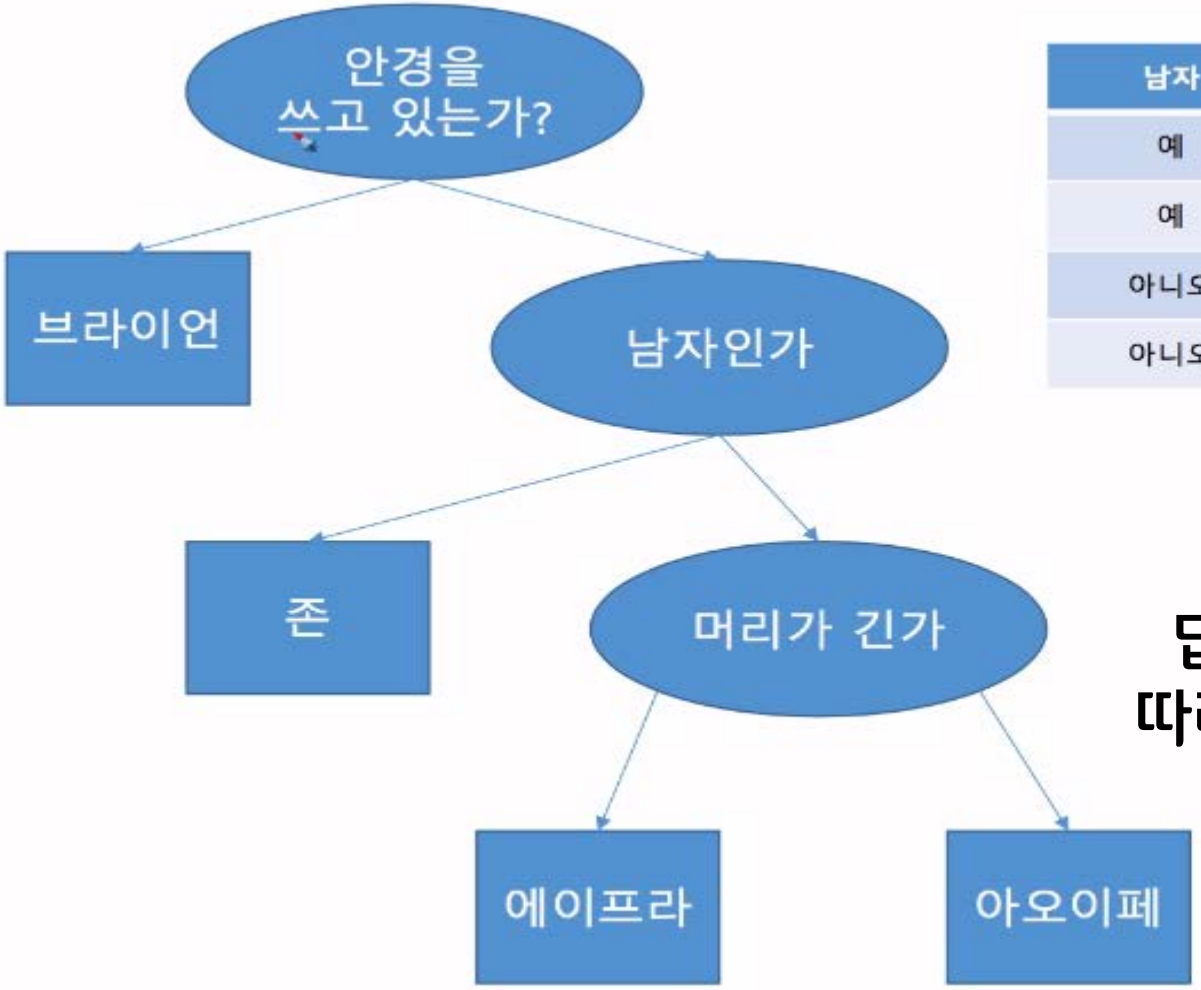
Making a Decision Tree

- 어떤 질문이 가장 많은 해답을 줄 것인가?
- -> 어떤 질문이 답의 모호성을 줄여줄 것인가?
- 문제를 통해서 Splitting Point를 설정

-> 남은 정보로 Splitting Point를 설정하는 식

!문제!
남자 농구선수를 분류를 한다고 하면
어떤 정보를 가장 먼저 Splitting
point로 설정하여야 할까?

Unit 01 | Decision Tree OverView



남자	긴 머리	안경	이름
예	아니오	예	브라이언
예	아니오	아니오	존
아니오	예	아니오	에이프라
아니오	아니오	아니오	아이오페

질문을 하나씩 할 때마다 답이 딱딱 떨어진다.
답을 명확하게 찾을 수 있어 모호성이 떨어진다!
따라서 이 Tree가 좀 더 좋은 Tree라고 할 수 있다.

contents

What is Entropy?

Entropy는 목적 달성을 위한 경우의 수를 정량적으로 표현하는 수치

-> Entropy 작을수록 목적 달성을 위한 경우의 수가 적어 얻을 수 있는 정보가 명확하다.

High Entropy -> Higher Uncertainty & Lower Entropy -> Lower Uncertainty

50명이 수학여행을 간다고 하면

Case A

50명

Lower Entropy

Case B

10명

10명

10명

10명

5명

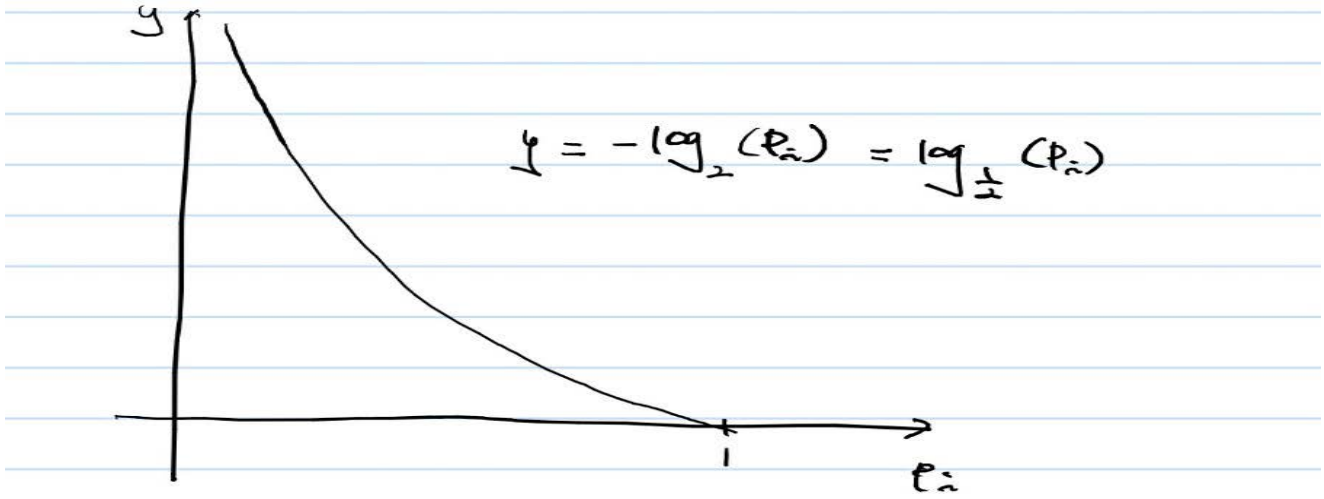
Higher Entropy

5명

Unit 02 | The algorithm of growing DT

What is Entropy?

$$h(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \qquad \text{where} \begin{cases} D & \text{Data set} \\ p_i & \text{Probability of label } i \end{cases}$$



Case 1) $P_1 = 1, P_2 = 0 \Rightarrow -\sum_{i=1}^{2} P_i \log_2(P_i) = 0$ Bits

⇒ lower Entrophy

Case 2) $P_1 = \frac{1}{2}, P_2 = \frac{1}{2} \Rightarrow -\sum_{i=1}^{2} P_i \log_2(P_i) = \frac{1}{2} + \frac{1}{2} = 1$ Bits

⇒ Higher Entrophy

데이터로 해볼까요?

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
2	middle_aged	high	no	fair	yes
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
6	middle_aged	low	yes	excellent	yes
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes
13	senior	medium	no	excellent	no

어떤 사람이 컴퓨터를 샀는지 분류해보자!

Age, Income, student, Credit_Rating : **Feature**

Class_buys_computer : **Label**

Label Entropy

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
2	middle_aged	high	no	fair	yes
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
6	middle_aged	low	yes	excellent	yes
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes
13	senior	medium	no	excellent	no

- Label에 대한 Entropy를 구해보자.

No → 1 $P_1 = 5/14$

Yes → 2 $P_2 = 9/14$

$$\text{Entropy}_{\text{label}} = -\sum_{i=1}^{2} P_i \log_2(P_i)$$

$$= -\frac{5}{14}\log\left(\frac{5}{14}\right) - \frac{9}{14}\log\left(\frac{9}{14}\right)$$

$$= 0.940$$

Growing a Decision Tree

- DT를 성장시키는 알고리즘이 필요! -> ID3(Information Gain) & CART(Gini index)
- Data의 Feature를 기준으로 분기를 진행.
- 어떤 Feature를 기준으로 가장 명확해 질까? -> Entropy가 작은 Feature
- 지속적인 분기를 어떻게 진행할까?

DT를 성장시키는 알고리즘

Feature "A"에 대해

(A의 Class는 3개이다.)

1. ID3 -> C4

- Information Gain

$$Gain(A) = Info(D) - Info_A(D_i)$$

$$Info(D) = Entropy_{label}$$

$$Info_A(D_i) = -\sum_{j=1}^{3} \frac{|D_j|}{|D|} * Entropy_{label_j}$$

2. CART

- Gini Index

$$Gini(A) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^{3} P_j \quad , P_j = \frac{|D_j|}{|D|}$$

Information Gain

1. ID3 -> C4

- Information Gain

$$Gain(A) = Info(D) - Info_A(D_i)$$

$$Info(D) = Entropy_{label}$$

$$Info_A(D_i) = -\sum_{j=1}^{3} \frac{|D_j|}{|D|} * Entropy_{label_j}$$

- Entropy를 도입하여 Branch Split을 해보자!
- Information Gain : Entropy를 활용하여 class 별 분류 시 Impurity를 측정하는 지표
- Gain = 전체 Entropy – 속성별 Entropy
- Gain이 높을 수록 명확한 정보를 얻을 수 있다.

ID3 Process

```
if 데이터 집합에 있는 모든 아이템이 같은 라벨임:
  분류 항목 표시를 반환함 (ex: buy_yes)
else:
  Find Best Spllit_branch_attribute(ex: attribute-age)
  해당 Attribute를 기준으로 dataset 분할
  Branch node 생성
    for each Branch
      brach_node.add(Recursive branch split)
  return branch node
```

ID3 Process

Information Gain - Age를 구해봅시다.

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
2	middle_aged	high	no	fair	yes
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
6	middle_aged	low	yes	excellent	yes
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes
13	senior	medium	no	excellent	no

- $Gain(age) = info_D(D) - info_{age}(D_i)$ $i = 1,2,3$

$info_D(D) = 0.94$

$info(age) = \sum_{i=1}^{3} \frac{|D_i|}{|D|} Entropy_{label_i}$

youth → 1, middle_aged → 2, senior → 3

- $\frac{|D_1|}{|D|} Entropy_{label_1} = \frac{5}{14}\left(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}\right)$
- $\frac{|D_2|}{|D|} Entropy_{label_2} = \frac{4}{14}\left(-\frac{4}{4}\log\frac{4}{4}\right)$
- $\frac{|D_3|}{|D|} Entropy_{label_3} = \frac{5}{14}\left(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}\right)$

$\therefore info(age) = 0.69$

$\therefore Gain(age) = 0.94 - 0.69 = 0.25$

Information Gain – 모든 Feature에 대한 Gain은?

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
2	middle_aged	high	no	fair	yes
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
6	middle_aged	low	yes	excellent	yes
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes
13	senior	medium	no	excellent	no

Gain(age) = 0.25

Gain(income) = 0.029

Gain(credit_rating) = 0.15

Gain(class_buys_computer) = 0.04

=> 최초 Split은 "age"로 진행.

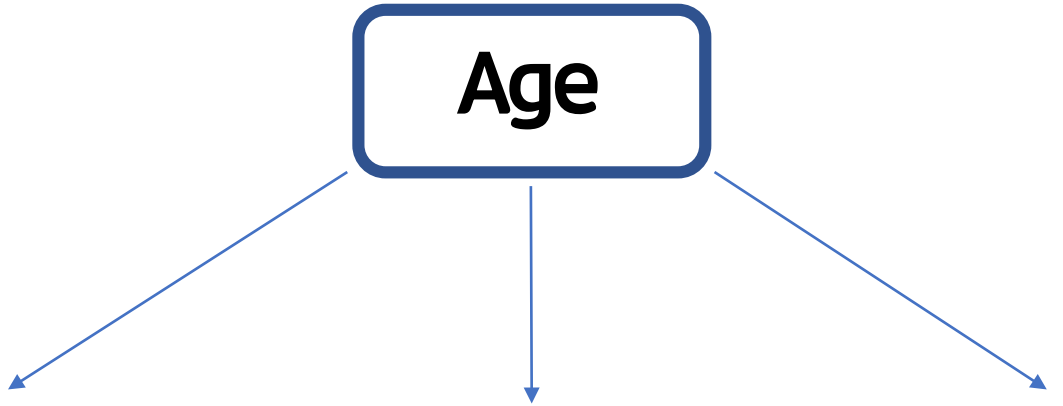
ID3 Process

```
if 데이터 집합에 있는 모든 아이템이 같은 라벨임:
  분류 항목 표시를 반환함 (ex: buy_yes)
else:
  Find Best Spllit_branch_attribute(ex: attribute-age)
  해당 Attribute를 기준으로 dataset 분할
  Branch node 생성
    for each Branch
      brach_node.add(Recursive branch split)
  return branch node
```

우리 여기까지 한거임

ID3 Process

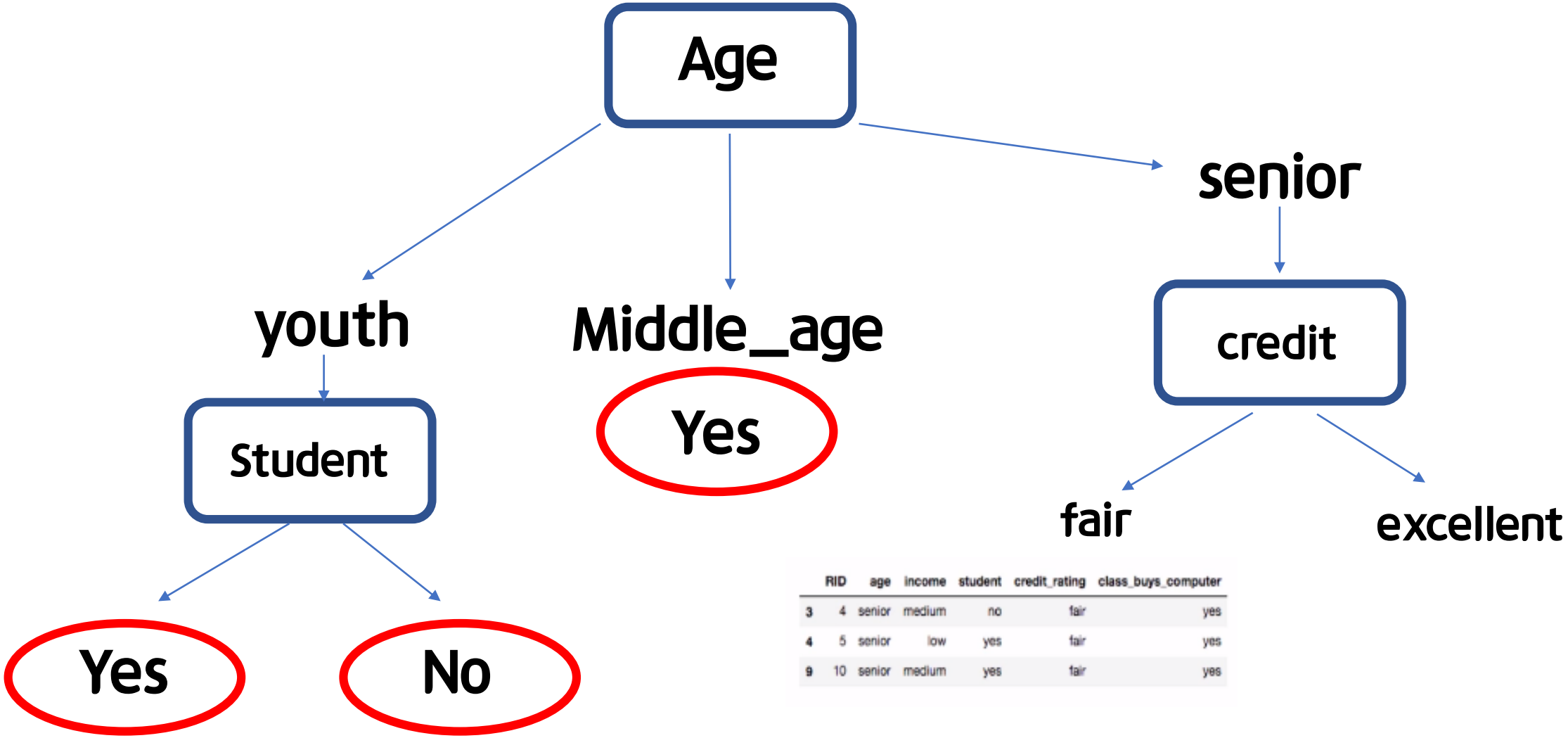
Unit 02 | The algorithm of growing DT



	RID	age	income	student	credit_rating	class_buys_computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
7	8	youth	medium	no	fair	no
8	9	youth	low	yes	fair	yes
10	11	youth	medium	yes	excellent	yes

	RID	age	income	student	credit_rating	class_buys_computer
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
5	6	senior	low	yes	excellent	no
9	10	senior	medium	yes	fair	yes
13	14	senior	medium	no	excellent	no

Unit 02 | The algorithm of growing DT



	RID	age	income	student	credit_rating	class_buys_computer
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
9	10	senior	medium	yes	fair	yes

Gini Index

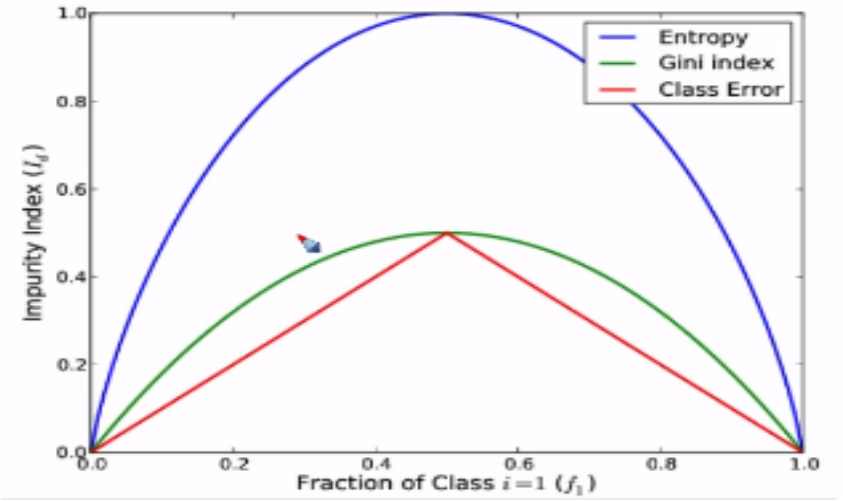
2. CART

- Gini Index

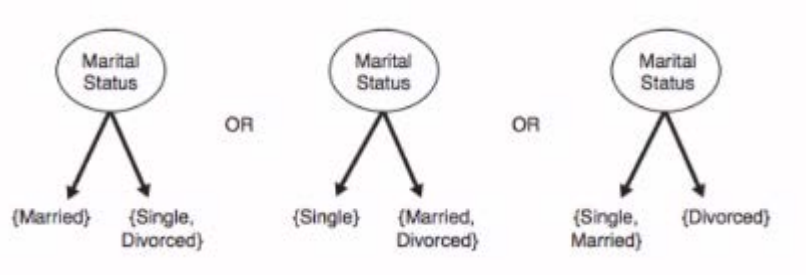
$$Gini(A) = \sum_{j=1}^{2} \frac{|D_j|}{|D|} * Gini(D_i)$$

$$Gini(D_i) = 1 - \sum_{j=1}^{3} P_j$$

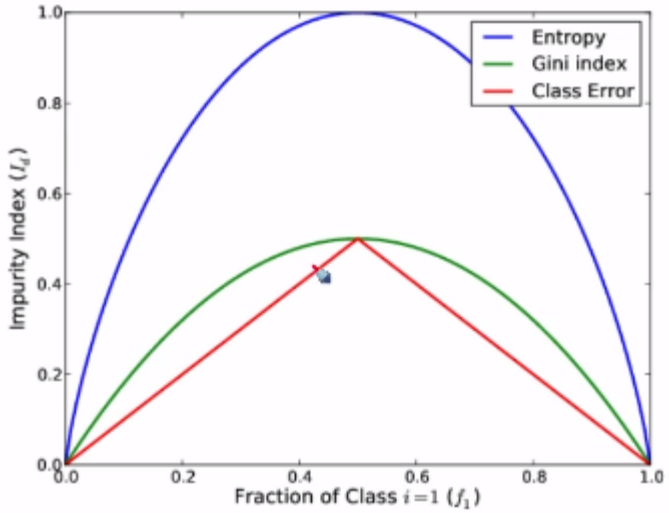
- CART 알고리즘의 Branch split
- Train 데이터를 나누었을 때 불순한 정도
- 데이터의 대상 속성을 얼마나 잘못 분류할지 계산
- Gini Index는 Entropy와 비슷한 그래프가 그려짐



Binary Split



- CART 알고리즘은 Binary split을 전제로 분석
- Feature의 데이터 분류의 개수가 k개 일 때 : $2^{k-1}-1$개 만큼의 Split 생성
- 데이터의 대상 속성을 얼마나 잘못 분류할지 계산
- Gini Index는 Entropy와 비슷한 그래프가 그려짐



Gini Index

	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
2	middle_aged	high	no	fair	yes
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
6	middle_aged	low	yes	excellent	yes
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes
13	senior	medium	no	excellent	no

Age $\mathrm{Gini}_{age}(D)$

Credit $\mathrm{Gini}_{credit}(D)$

Income $\mathrm{Gini}_{income}(D)$

Student $\mathrm{Gini}_{student}(D)$

Gini Index

Age

$$\text{Gini}_{age}(D)\ age \in \{youth, middle\_age, senior\}$$

$$age_1 \in \{youth\} = \{middle\_age, senior\}$$
$$age_2 \in \{middle\_age\} = \{youth, senior\}$$
$$age_3 \in \{senior\} = \{youth, middle\_age\}$$

	RID	age	income	student	credit_rating	class_buys_computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
7	8	youth	medium	no	fair	no
8	9	youth	low	yes	fair	yes
10	11	youth	medium	yes	excellent	yes

	RID	age	income	student	credit_rating	class_buys_computer
2	3	middle_aged	high	no	fair	yes
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
5	6	senior	low	yes	excellent	no
6	7	middle_aged	low	yes	excellent	yes
9	10	senior	medium	yes	fair	yes
11	12	middle_aged	medium	no	excellent	yes
12	13	middle_aged	high	yes	fair	yes
13	14	senior	medium	no	excellent	no

Gini Index

	RID	age	income	student	credit_rating	class_buys_computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
7	8	youth	medium	no	fair	no
8	9	youth	low	yes	fair	yes
10	11	youth	medium	yes	excellent	yes

	RID	age	income	student	credit_rating	class_buys_computer
2	3	middle_aged	high	no	fair	yes
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
5	6	senior	low	yes	excellent	no
6	7	middle_aged	low	yes	excellent	yes
9	10	senior	medium	yes	fair	yes
11	12	middle_aged	medium	no	excellent	yes
12	13	middle_aged	high	yes	fair	yes
13	14	senior	medium	no	excellent	no

① {youth}, {middle_aged, Senior}

② {middle_aged}, {youth, Senior}

③ {Senior}, {youth, middle_aged}

$Gini ① \Rightarrow \sum_{j=1}^{2} \frac{|D_j|}{|D|} * Gini(D_j) = \frac{5}{14} Gini(D_1) + \frac{9}{14} Gini(D_2)$

$Gini(D_1) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$

$Gini(D_2) = 1 - \left(\frac{2}{9}\right)^2 - \left(\frac{7}{9}\right)^2$

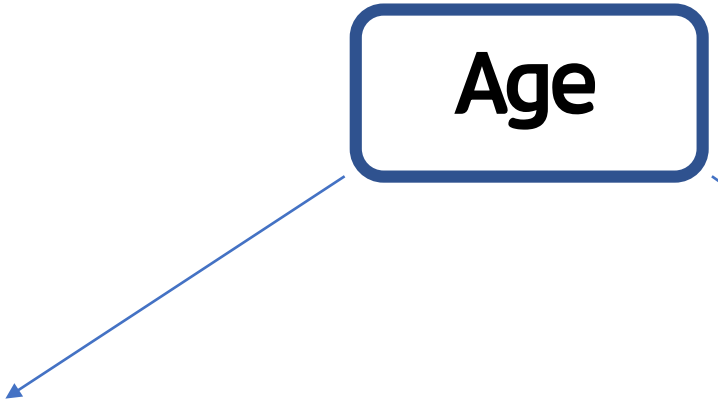
⇒ ②, ③도 같은 방식으로 구해 가장작은 값을 Gini(age)로 선정.

∴ $Min(①, ②, ③) = Gini(age)$

Unit 02 | The algorithm of growing DT

Gini Index

$$Min(Gini_{age_i}) = 0.357$$
$$Min(Gini_{income_i}) = 0.443$$
$$Min(Gini_{credit}) = 0.429$$
$$Min(Gini_{student}) = 0.367$$



Middle_aged

	age	income	student	credit_rating	class_buys_computer
2	middle_aged	high	no	fair	yes
6	middle_aged	low	yes	excellent	yes
11	middle_aged	medium	no	excellent	yes
12	middle_aged	high	yes	fair	yes

Youth,senior

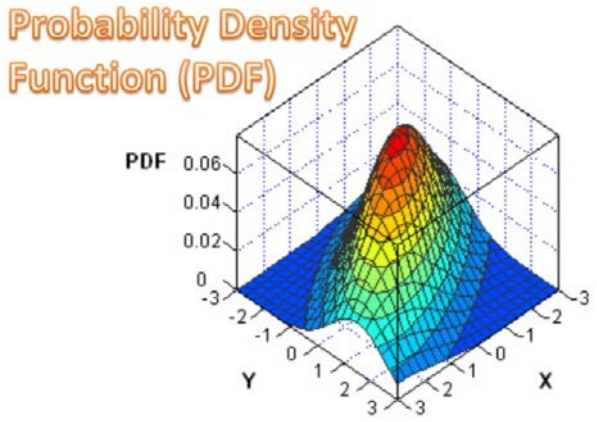
	age	income	student	credit_rating	class_buys_computer
0	youth	high	no	fair	no
1	youth	high	no	excellent	no
3	senior	medium	no	fair	yes
4	senior	low	yes	fair	yes
5	senior	low	yes	excellent	no
7	youth	medium	no	fair	no
8	youth	low	yes	fair	yes
9	senior	medium	yes	fair	yes
10	youth	medium	yes	excellent	yes
13	senior	medium	no	excellent	no

Continuous Feature 나누는 방법

1) 전체 데이터를 모두 기준점으로 한다.

2) 중위수, 4분위수들을 기준점으로 한다.

3) Y-class 값이 바뀌는 수를 기준점으로 한다.



Continuous Feature 나누는 방법

Step1. Split할 Feature를 Sorting 시킨다.

	ID	STREAM	SLOPE	ELEVATION	VEGETATION
0	1	False	steep	3900	chapparal
1	2	True	moderate	300	riparian
2	3	True	steep	1500	riparian
3	4	False	steep	1200	chapparal
4	5	False	flat	4450	conifer
5	6	True	steep	5000	conifer
6	7	True	steep	3000	chapparal

Sorting

	ID	STREAM	SLOPE	ELEVATION	VEGETATION
0	1	False	steep	3900	chapparal
1	2	True	moderate	300	riparian
2	3	True	steep	1500	riparian
3	4	False	steep	1200	chapparal
4	5	False	flat	4450	conifer
5	6	True	steep	5000	conifer
6	7	True	steep	3000	chapparal

Continuous Feature 나누는 방법

Step2. Y-class 데이터가 변경되는 부분들을 찾는다.

	ID	STREAM	SLOPE	ELEVATION	VEGETATION	
1	2	True	moderate	300	riparian	(1)
3	4	False	steep	1200	chapparal	(2)
2	3	True	steep	1500	riparian	(3)
6	7	True	steep	3000	chapparal	
0	1	False	steep	3900	chapparal	(4)
4	5	False	flat	4450	conifer	
5	6	True	steep	5000	conifer	

Step3. 구간별 경계 평균값으로 기준을 잡는다.

	ID	STREAM	SLOPE	ELEVATION	VEGETATION	
1	2	True	moderate	300	riparian	750
3	4	False	steep	1200	chapparal	1,350
2	3	True	steep	1500	riparian	2250
6	7	True	steep	3000	chapparal	
0	1	False	steep	3900	chapparal	4175
4	5	False	flat	4450	conifer	
5	6	True	steep	5000	conifer	

Continuous Feature 나누는 방법

Step4. 구간별 경계값을 기준으로 Entropy or Gini를 산출한다.

$$\text{Gain}(elec_{750}) = Info(D) - Info_{elec_{750}}(D)$$
$$\text{Gain}(elec_{1350}) = Info(D) - Info_{elec_{1350}}(D)$$
$$\text{Gain}(elec_{2250}) = Info(D) - Info_{elec_{2250}}(D)$$
$$\text{Gain}(elec_{4175}) = Info(D) - Info_{elec_{4175}}(D)$$

$$Max(\text{Gain}(elec))$$

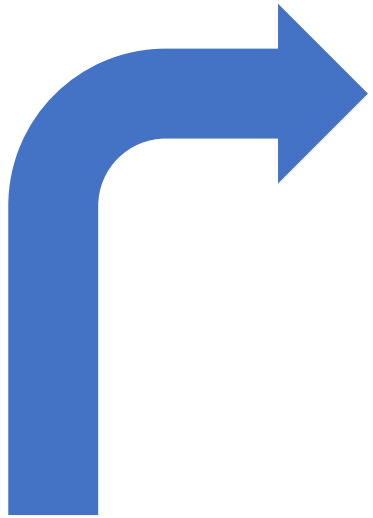
Continuous Feature 나누는 방법

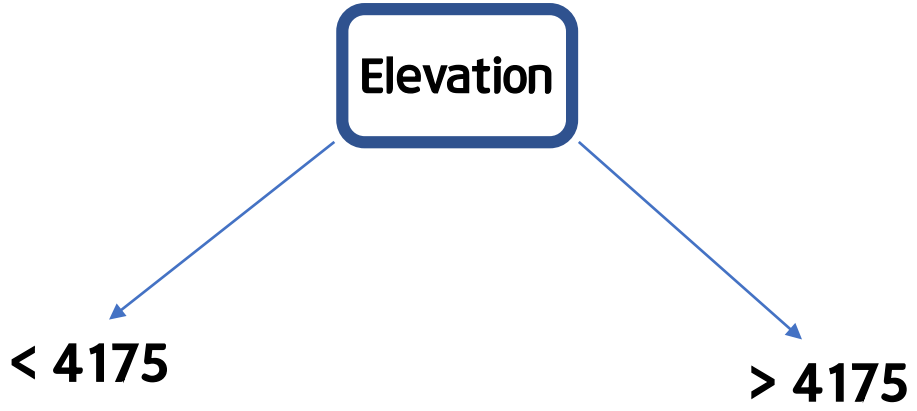
Step5. 최종 Split Point 선택

Stream — 0.3

Slope — 0.5

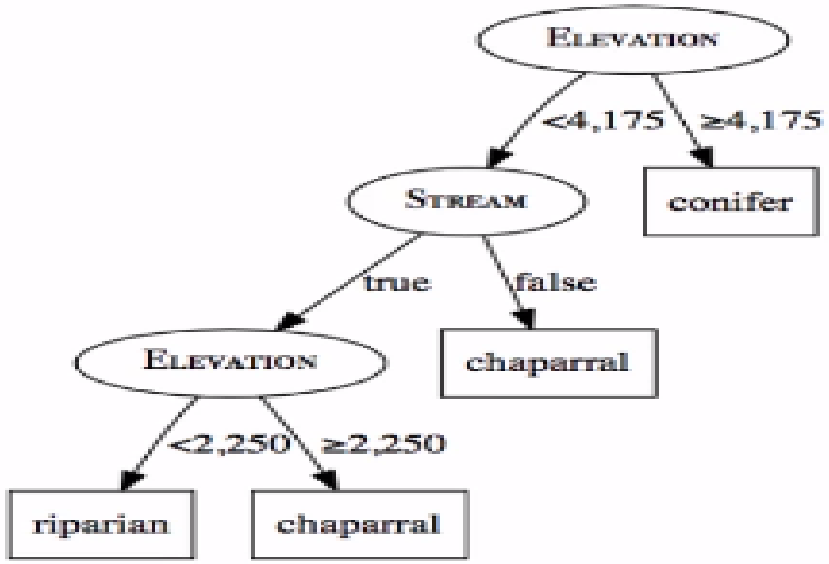
Elevation — 750 : 0.3
1350 : 0.18
2250 : 0.59
4175 : 0.86





Continuous Attribute Branch 특징

- 명목속성과 달리, 여러 번 재사용 가능 -> 경계값은 달라진다.
- 연속값과 명목값을 동시에 Split가능



Regression using Decision Tree

- Y – Continous일 경우

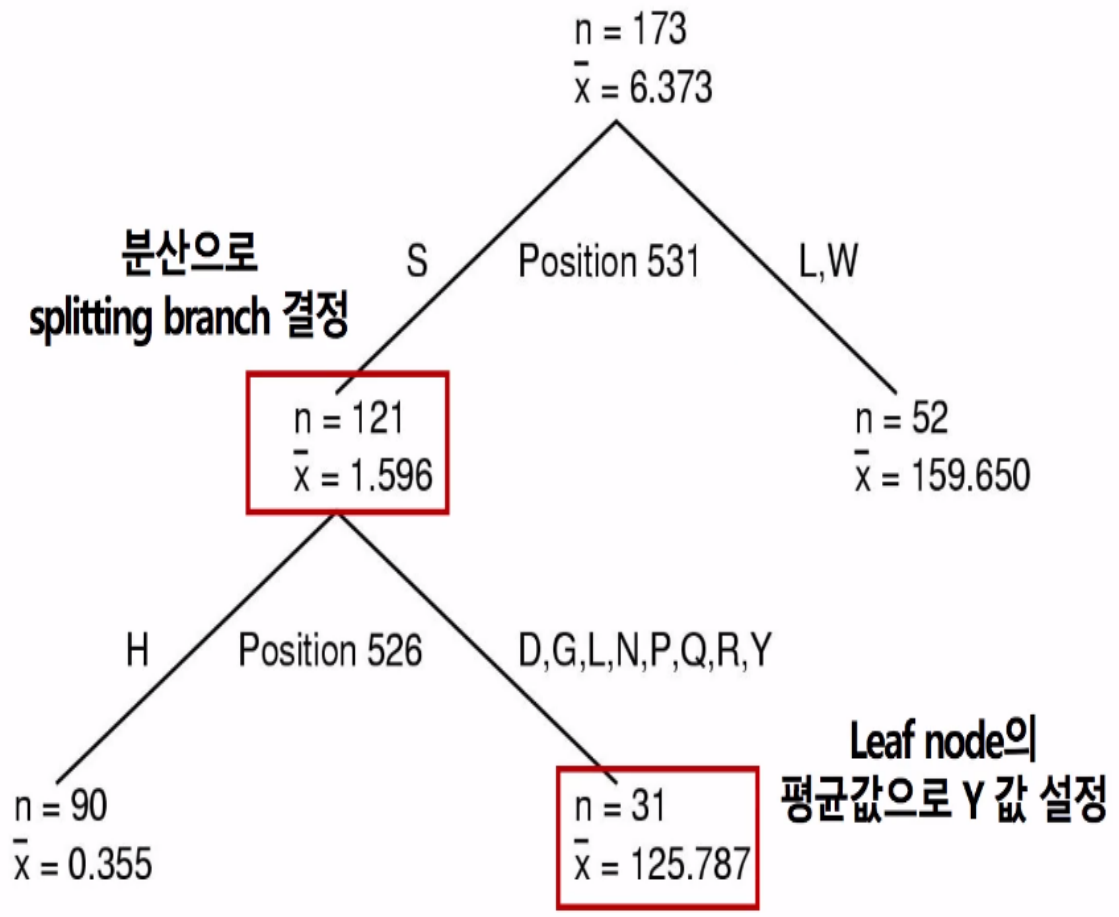
1) Split measure의 변화 : Entropy -> Variance

2) 예측 : 분류된 Instacne들의 평균

3) 나머진 동일하다!

$$var(D) = \frac{\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2}{n-1} \qquad \text{where } \bar{y} = \frac{\sum_{i=1}^{n} y^{(i)}}{n}$$

$$argmax \sum \frac{|D_i|}{|D|} * Var(D_i)$$



Regression using Decision Tree

season	weekday	weathersit	cnt
2	6	2	352
4	1	2	109
2	4	1	421
4	6	1	165
2	5	1	12
3	5	1	161
2	2	3	162
1	1	3	79
2	2	1	112

Season

$$WeightVar(Season) = \frac{1}{9} * (79-79)^2 + \frac{5}{9} * \frac{(352-211.8)^2+(421-211.8)^2+(12-211.8)^2+(162-211.8)^2+(112-211.8)^2}{4} + \frac{1}{9} * (161-161)^2 + \frac{2}{9} * \frac{(109-137)^2+(165-137)^2}{1} = 16429.1$$

Weekday

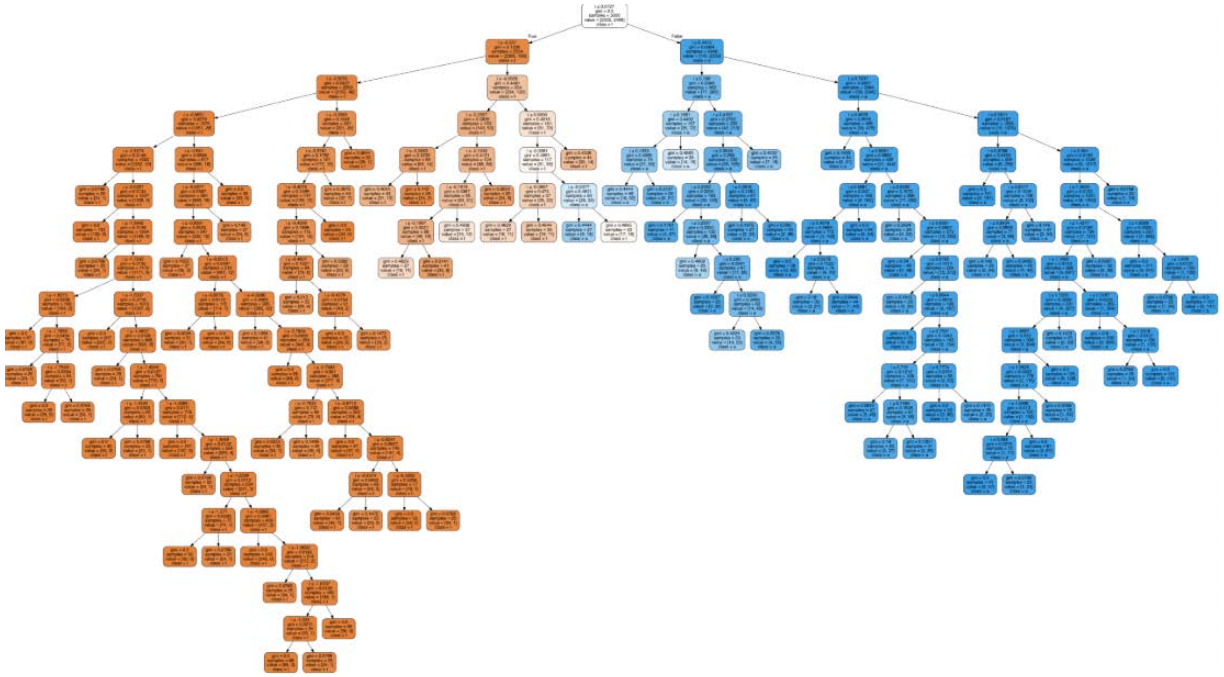
$$WeightVar(Weekday) = \frac{2}{9} * \frac{(109-94)^2+(79-94)^2}{1} + \frac{2}{9} * \frac{(162-137)^2+(112-137)^2}{1} + \frac{1}{9} * (421-421)^2 + \frac{2}{9} * \frac{(161-86.5)^2+(12-86.5)^2}{1} + \frac{2}{9} * \frac{(352-258.5)^2+(165-258.5)^2}{1} = 6730$$

Weathersit

$$WeightVar(Weathersit) = \frac{4}{9} * \frac{(421-174.2)^2+(165-174.2)^2+(12-174.2)^2+(161-174.2)^2+(112-174.2)^2}{4} + \frac{2}{9} * \frac{(352-230.5)^2+(109-230.5)^2}{1} + \frac{2}{9} * \frac{(79-120.5)^2+(112-120.5)^2}{1} = 19646.83$$

contents

DT 문제점



- Class의 leaf node의 결정
- 너무 많을 경우 -> Overfitting
- 노드의 데이터가 1개 -> 어떤 지점에서 트리 가지치기를 해야할 지 결정해야 함

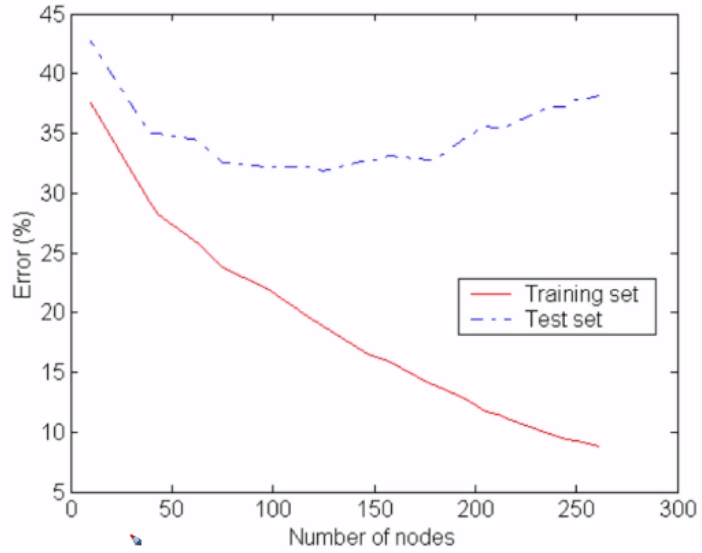
Pruning

- Tree를 생성할 때, 일정 기준 이하면 노드를 생성하지 않는다.

-> 하위 노드의 개수 or 하드 노드의 비율을 Threshold로 잡는다.

- Tree를 생성한 후, 트리의 오차율 최소화를 위해 실행
- 검증을 위한 Validation Set을 따로 생성

오분류율 최소화



DT 특징

- Top-Down, Recursive, Divide and Conquer 기법
- Greedy 알고리즘 – 부분 최적화
- 확실한 정보의 선택기준은 알고리즘 별로 차이가 남(entropy, gini – 하이퍼파라미터)
- Tree 생성 후 Pruning을 통해 Tree generalization 시행

DT 장점

- 해석의 용이성
- 다른 알고리즘과 비교하여 Feature importance를 구할 수 있는 매우 큰 장점이 있다.
- Underfitting, Overfitting 되기 쉽다. -> 왜 장점일까?

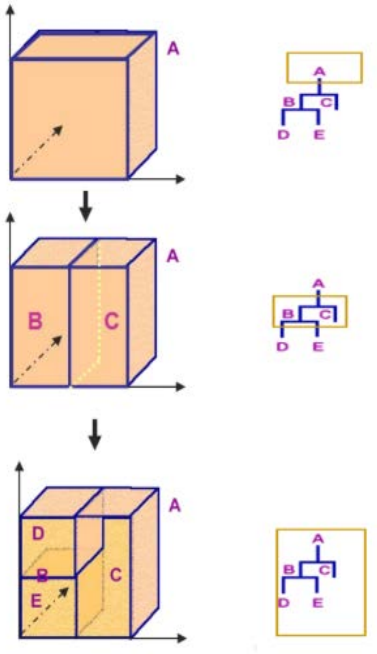
DT 특징 - 단점

의사결정 나무는 결정경계가 데이터 축에 수직이기 때문에 비선형 데이터 분류에 적합하지 않다.

또한, Depth가 길어지면 Overfitting 되기 쉽다.

그.래.서. Ensemble 기법을 통해 여러 개의 의사결정나무를 만들어 그 결과를 종합해 이러한 문제를 극복할 수 있다.

Ensemble2 발표에서 이러한 앙상블 기법을 다뤄 보겠습니다.
(Bagging, Boosting, Randomforest, AdaBoost, Gradient boosting, XGBboosting, GBM, LightGBM)



contents

Sklearn with Decision Tree

sklearn.tree.DecisionTreeClassifier

class sklearn.tree. **DecisionTreeClassifier** (*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False*) [source]

- Criterion : 트리를 성장시키는 알고리즘(Entropy or Gini)
- Max depth : 얼마나 깊게 DT를 성장 시킬 것이냐
- Min sample split : 최소한의 개수가 몇 개 이상이여야 split을 하겠냐
- Min sample leaf : 리프 노드에 있어야하는 샘플의 최소 수

Q & A

들어주셔서 감사합니다.

이제 과제를 설명 드리겠습니다.

과제 및 실습

Assignment 1 : wk3_DT_assignment1.ipynb를 켜주세요.

작성한 함수를 통해 구한 값으로 보여주세요!

문제1) income의 이진분류를 얻는 함수 get_binary_split(pd_data, "income")을 통해 보여주세요.

문제2) 가장 Gini계수가 높은 Feature 즉 분류를 하는데 가장 중요한 변수를 선정하시고 get_attribute_gini_index 함수를 통해 Gini index를 제시해주세요.

문제3) 2에서 구한 Feature로 DataFrame을 분류 해주시고 나눠진 2개의 클래스에서 각각 다음으로 중요한 Feature를 선정해주시고 Gini index를 제시해주세요.

과제 및 실습

Assignment 2 : wk3_DT_assignment2.ipynb를 켜주세요.

저번 로지스틱 때 사용한 타이타닉 코드입니다.

문제) pruning을 통해 성능을 높여보세요. 그리고 위의 default DT성능과 비교하여 Accuracy가 얼마나 개선되었는지 구해보세요.