Session 10

Decision Tree

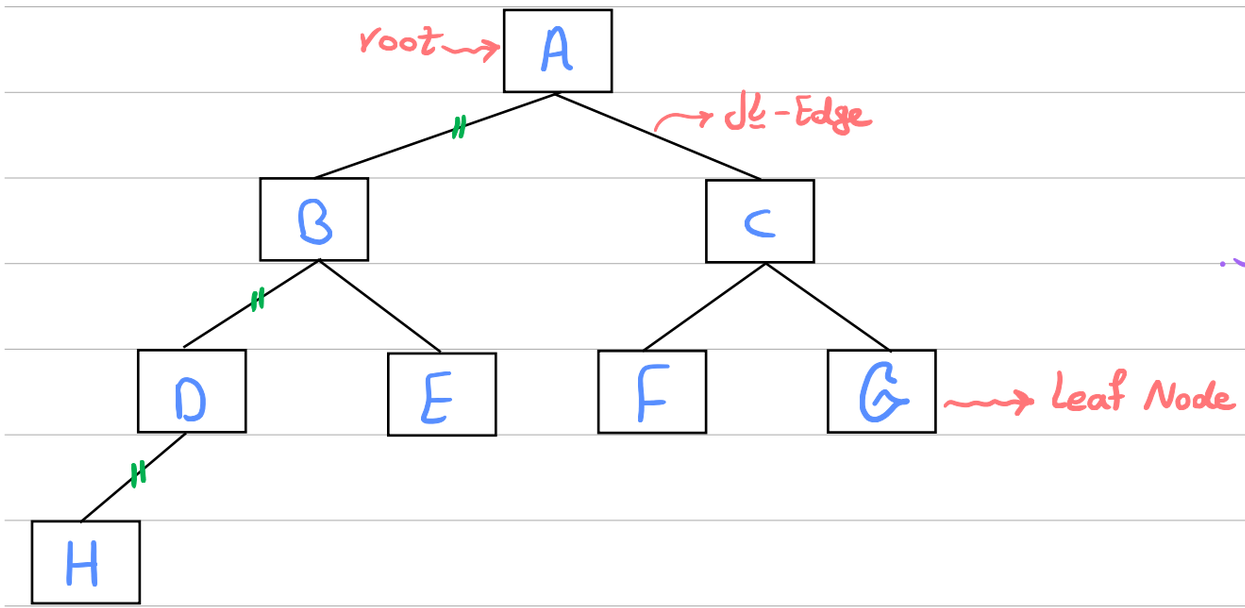
Machine Learning | Zahra Amini

Telegram: @zahraamini_ai & Instagram:@zahraamini_ai & LinkedIn: @zahraamini-ai

https://zil.ink/zahraamini

درخت تصمیم ← (DT) Decision Tree

DT یکی از قدیمی ترین روش های طبقه بندی ست.



ساختمان داده درخت:

عمق درخت ← تعداد تمام یالها از ریشه تا عمیق ترین برگ.

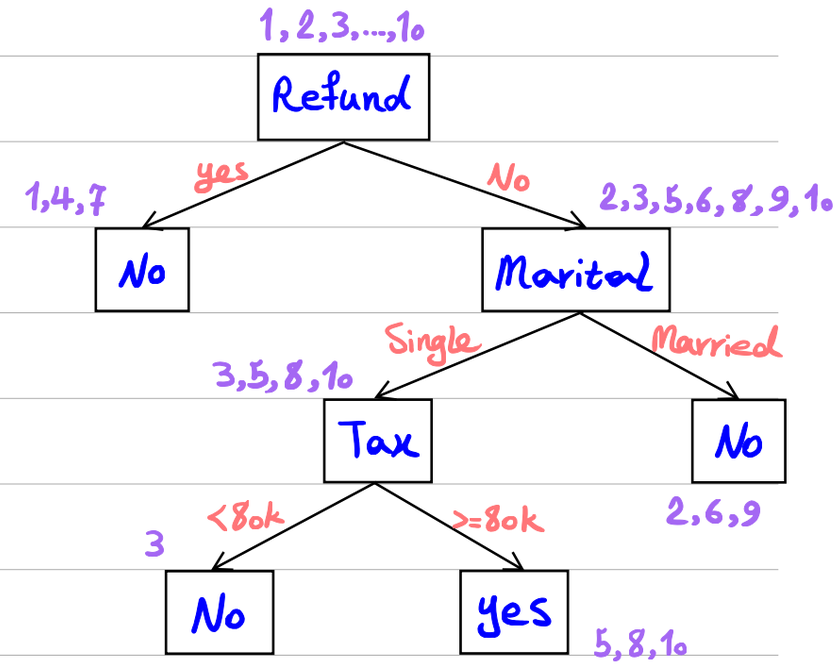
عمق درخت = 3

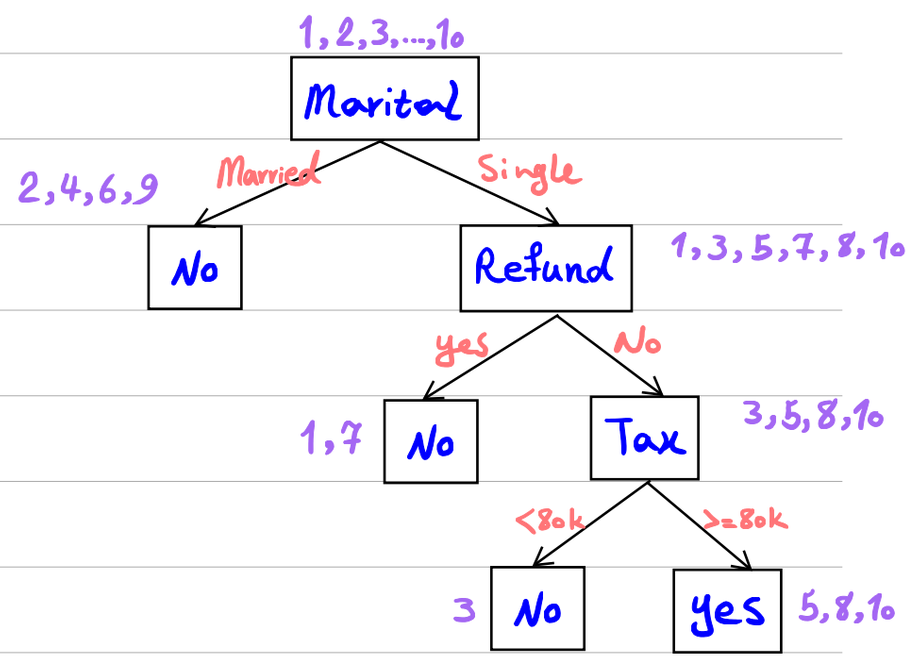
گره داخلی ← ویژگی

گره برگ ← class label

بازپرداخت وام قبلی — وضعیت تأهل — مالیات بر درآمد

	Refund	Marital Status	Taxable Income	Class
1	yes	Single	125k	No
2	No	Married	100k	No
3	No	Single	70k	No
4	yes	Married	120k	No
5	No	Single	95k	yes
6	No	Married	60k	No
7	yes	Single	220k	No
8	No	Single	85k	yes
9	No	Married	75k	No
10	No	Single	90k	yes





X^{test} : No, Married, 80k ⟶ Class=? No

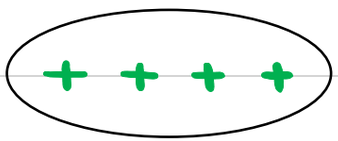
؟: کدام درخت بهتر است؟ از کدام ویژگی شروع کنیم؟

برای پاسخ به این سؤال نیاز به اطلاعات بیشتری داریم.

الگوریتم های DT اغلب بر پایه‌ی جستجو حریصانه کار می‌کنند.

$$\text{Entropy}(S) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

Entropy ← معیار ناخالصی یک گره ← Measure of Node Impurity

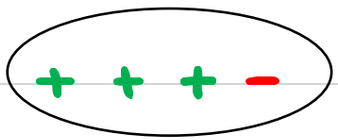


Entropy = 0

$P_+ = \frac{4}{4} = 1$, $P_- = \frac{0}{4} = 0$

$\text{Entropy}([4+, 0-]) =$

$-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$

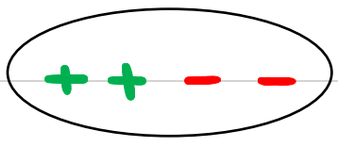


Entropy = 0.811

$P_+ = \frac{3}{4}$, $P_- = \frac{1}{4}$

$\text{Entropy}([3+, 1-]) =$

$-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.811$



Entropy = 1

$P_+ = \frac{2}{4}$, $P_- = \frac{2}{4}$

$\text{Entropy}([2+, 2-]) =$

$-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$

$0 \leq \text{Entropy} \leq 1$

* هرچقدر Entropy کمتر خلوص بیشتر (خالص‌تر)

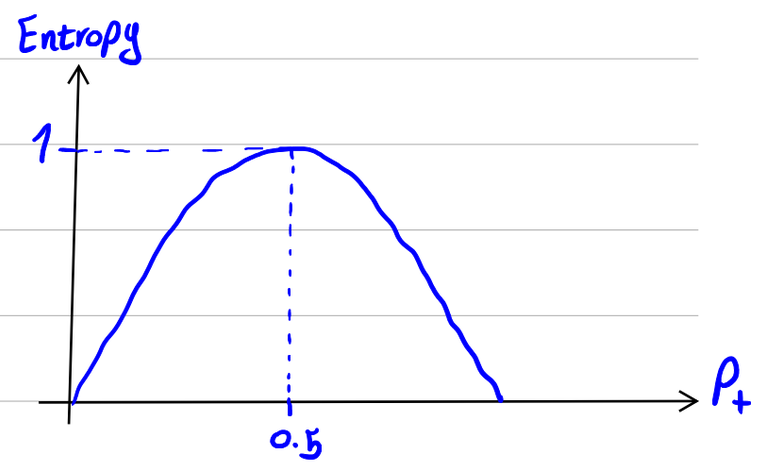
	Wind	Temp	OutLook	Humidity	Play?
1	Weak	Hot	Sunny	High	Play
2	Strong	Hot	Sunny	High	Play
3	Weak	Hot	Rain	High	Stay
4	Weak	Mid	Overcast	High	Play
5	Strong	Cold	Rain	Normal	Stay
6	Weak	Cold	Overcast	Normal	Play
7	Strong	Cold	Rain	Normal	Stay
8	Weak	Mid	Sunny	Normal	Play
9	Weak	Cold	Sunny	Normal	Play
10	Strong	Mid	Overcast	Normal	Play
11	Weak	Mid	Sunny	High	Stay
12	Strong	Mid	Rain	High	Stay
13	Weak	Hot	Overcast	Normal	Play
14	Weak	Cold	Rain	High	Play

$$\text{Entropy}(S) =$$

$$-\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) =$$

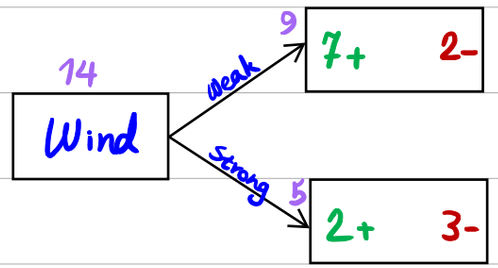
$$0.940$$

$$P_+ = \frac{9}{14} \qquad P_- = \frac{5}{14}$$



Gain ← بهره

$$Gain(S, A) = Ent(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Ent(S_v)$$



$$Ent(weak) = -\frac{7}{9}\log_2\frac{7}{9} - \frac{2}{9}\log_2\frac{2}{9} = 0.764$$

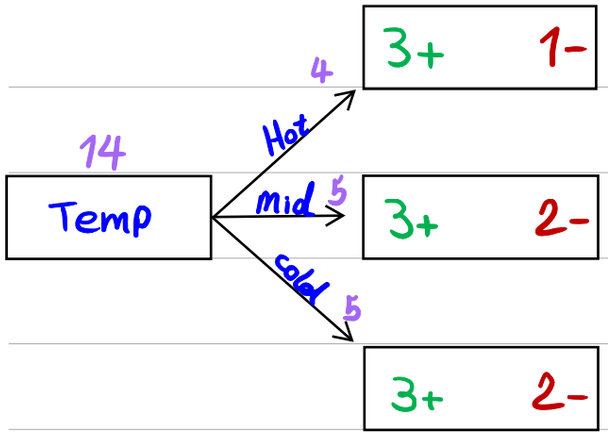
$$Ent(Strong) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.97$$

$$E(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94$$

$$Gain(D, Wind) = E(D) - \sum_{v \in value(wind)} \frac{|D_v|}{|D|} Ent(D_v) =$$

$$0.94 - \left(\frac{9}{14}\underbrace{Ent(weak)}_{0.764} + \frac{5}{14}\underbrace{Ent(strong)}_{0.97}\right) =$$

$$0.94 - (0.491 + 0.346) = 0.94 - 0.837 = 0.102$$



$$Ent(Hot) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.811$$

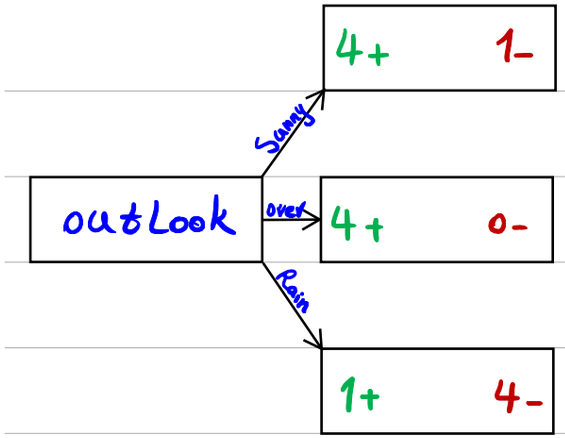
$$Ent(Mid) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) = 0.97 = E(cold)$$

$$Gain(D, Temp) = Ent(D) - \sum_{v \in Values(Temp)} \frac{|D_v|}{|D|} Ent(D_v) =$$

$$0.94 - \left(\frac{4}{14}\overset{0.811}{\cancel{Ent}}(Hot) + \frac{5}{14}\overset{0.97}{\cancel{Ent}}(Mid) + \frac{5}{14}\overset{0.97}{\cancel{Ent}}(cold)\right) =$$

$$0.94 - (0.231 + 0.346 + 0.346) = 0.94 - 0.932 =$$

$$0.008$$

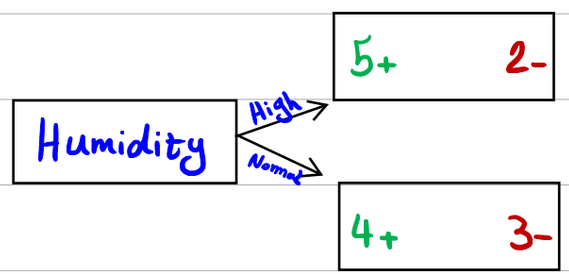


$$Ent(Sunny) = -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right) = 0.722$$

$$Ent(overcast) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log\frac{0}{4} = 0$$

$$Ent(Rain) = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right) = 0.722$$

$$Gain(O, outLook) = Ent(D) - \left(\frac{5}{14}\underbrace{Ent(S)}_{0.722} + \frac{4}{14}\underbrace{Ent(O)}_{0} + \frac{5}{14}\underbrace{Ent(R)}_{0.722}\right) = 0.94 - (0.258 + 0 + 0.258) = 0.94 - 0.516 = 0.424$$



$$Ent(High) = -\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7} = 0.863$$

$$Ent(Normal) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.985$$

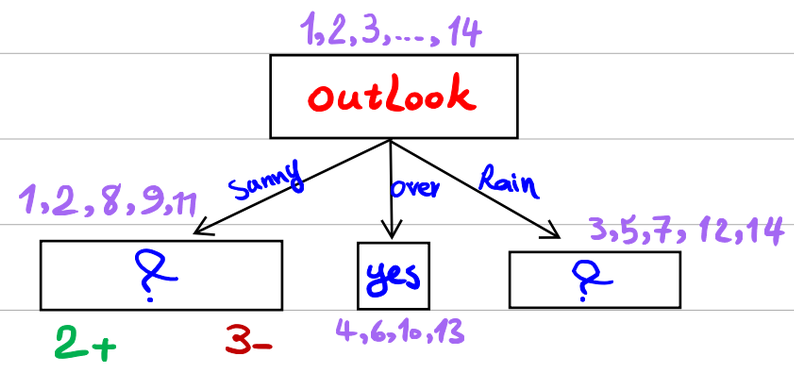
$$Gain(O, Humidity) = Ent(D) - \left(\frac{7}{14}\underbrace{Ent(H)}_{0.863} + \frac{7}{14}\underbrace{Ent(N)}_{0.985}\right) = 0.94 - (0.431 + 0.493) = 0.94 - 0.924 = 0.016$$

؟: ما Gain تمام feature ها را به دست آوردیم. کدام feature را براساس Gain می‌توان انتخاب کرد؟

Gain
- Wind → 0.102
- Temp → 0.008
- OutLook → 0.424
- Humidity → 0.016

$\xrightarrow{\text{Max}}$ OutLook

ما به دنبال خلوص بیشتر هستیم و Gain بیشتر یعنی خلوص بیشتر.



$D = \{1, 2, 8, 9, 11\}$

$$Gain(D, Humidity) = Ent(D) - \sum \frac{|D_v|}{|D|} Ent(D_v) = 0.97 - \frac{3}{5} \times Ent(H) - \frac{2}{5} Ent(N) = 0.97 - 0 - 0 = 0.97$$

$$Gain(D, Temp) = Ent(D) - \frac{2}{5} Ent(H) - \frac{2}{5} Ent(m) - \frac{1}{5} Ent(c) = 0.97 - 0 - 0.4 - 0 = 0.57$$

$$Gain(D, wind) = \overset{0.97}{\cancel{Ent}}(D) - \frac{2}{5} \overset{1}{\cancel{Ent}}(s) - \frac{3}{5} \overset{0.918}{\cancel{Ent}}(w) = 0.019$$

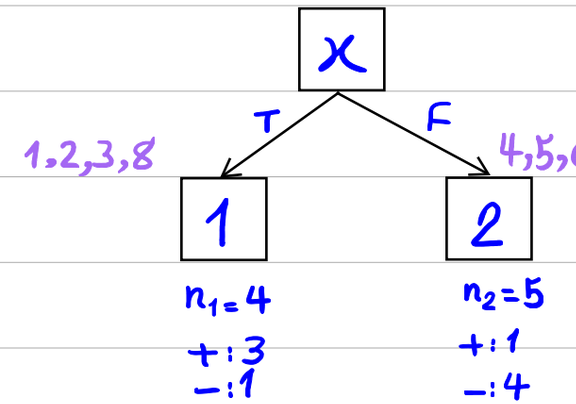
$\xrightarrow{\text{max}}$ Humidity

ضریب Gini:

$$Gini(t) = 1 - \sum_{i=1}^{c} P_i^2$$

$$GiniSplit(A) = \sum_{i=1}^{k} \frac{n_i}{n} Gini(i)$$

	x	y	class
1	T	T	+
2	T	T	+
3	T	F	−
4	F	F	+
5	F	T	−
6	F	T	−
7	F	F	−
8	T	F	+
9	F	T	−

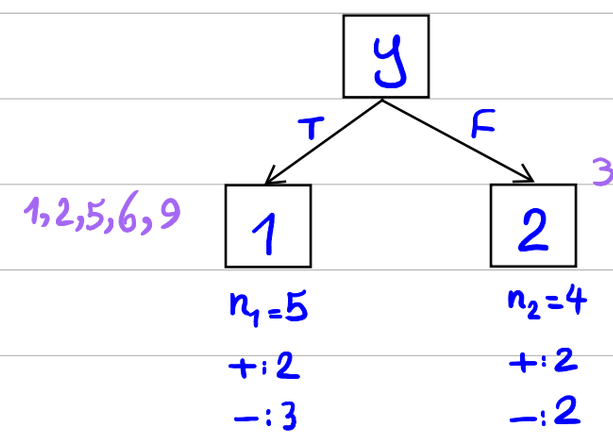


$$Gini(1) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = 1 - \left(\frac{9}{16} + \frac{1}{16}\right) = \frac{6}{16}$$

$$Gini(2) = 1 - \left(\left(\frac{1}{5}\right)^2 + \left(\frac{4}{5}\right)^2\right) = 1 - \left(\frac{1}{25} + \frac{16}{25}\right) = \frac{8}{25}$$

$$GiniSplit(x) = \frac{n_1}{n} Gini(1) + \frac{n_2}{n} Gini(2) =$$

$$\frac{4}{9} \times \frac{6}{16} + \frac{5}{9} \times \frac{8}{25} = 0.166 + 0.177 = \underline{0.343}$$



$$Gini(1) = 1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right) = 1 - \left(\frac{9}{25} + \frac{4}{25}\right) = \frac{12}{25}$$

$$Gini(2) = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 1 - \left(\frac{4}{16} + \frac{4}{16}\right) = \frac{8}{16}$$

$x \xleftarrow{min}$

$$GiniSplit(y) = \frac{n_1}{n} Gini(1) + \frac{n_2}{n} Gini(2) =$$

هرچه دنبال Gini کم هستیم.

$$\frac{5}{9} \times \frac{12}{25} + \frac{4}{9} \times \frac{8}{16} = 0.222 + 0.266 = 0.488$$

ج: اگر ویژگی مان چند مقداری باشد، از دو انشعاب استفاده کنیم یا چند انشعاب؟

Car Type	Class
Family	C1
Sport	C1
Luxury	C1
Sport	C1
Family	C2
Family	C2
Family	C2
Family	C2
Sport	C2
Luxury	C2

class	Car Type		
	a Family	b Sport	c Luxury
C1	1	2	1
C2	4	1	1
	5	3	2

$$GiniSplit = \frac{5}{10} \times Gini(a) + \frac{3}{10} \times Gini(b) + \frac{2}{10} \times Gini(c)$$

$$= 0.392$$

$$Gini(a) = 1 - \left(\left(\frac{1}{5}\right)^2 + \left(\frac{4}{5}\right)^2\right) = 0.32$$

$$Gini(b) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.44$$

$$Gini(c) = 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0.5$$

class	Car Type	
	Sport, Luxury	Family
C1	a 3	b 1
C2	2	4
	5	5

$$GiniSplit = \frac{5}{10} \times Gini(a) + \frac{5}{10} \times Gini(b)$$

$$= 0.4$$

$$Gini(a) = 1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right) = 0.48$$

$$Gini(b) = 1 - \left(\left(\frac{1}{5}\right)^2 + \left(\frac{4}{5}\right)^2\right) = 0.32$$

class	Car Type	
	Sport	Luxury, Family
C1	a 2	b 2
C2	1	5
	3	7

$$GiniSplit = \frac{3}{10} \times Gini(a) + \frac{7}{10} Gini(a) = 0.417$$

$$Gini(a) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.44$$

$$Gini(b) = 1 - \left(\left(\frac{2}{7}\right)^2 + \left(\frac{5}{7}\right)^2\right) = 0.408$$

کمترین Gini برای چند انشعاب است.

$$0 \leq Gini \leq 0.5$$

نوبتی هم باشه، نوبت شماست :)

RID	Age	Income	Student	Credit_rating	Class: buys computer
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	middle_aged	High	No	Fair	Yes
4	Senior	Medium	No	Fair	Yes
5	Senior	Low	Yes	Fair	Yes
6	Senior	Low	Yes	Excellent	No
7	middle_aged	Low	Yes	Excellent	Yes
8	Youth	Medium	No	Fair	No
9	Youth	Low	Yes	Fair	Yes
10	Senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	middle_aged	Medium	No	Excellent	Yes
13	middle_aged	High	Yes	Fair	Yes
14	Senior	Medium	No	Excellent	No

کدام ویژگی را برای ریشه درخت انتخاب کنیم؟

- از معیار Gain استفاده کنید.

Gain (age) = ?

Gain (income) = ?

Gain (student) = ?

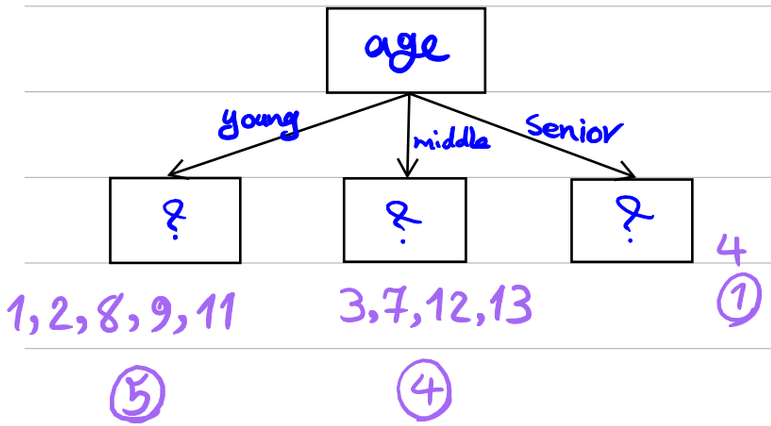
Gain (credit_rating) = ?

+ −

9 5

Gain (age) =

$$\text{Entropy}(S) = \sum_{i=1}^{c} -P_i \log_2 P_i$$



$$\text{Ent}(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

$$\text{Gain}(S, A) = \text{Ent}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|}\,\text{Ent}(S_v)$$

$$0.94 - \frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694$$

بازپرداخت وام قبلی وضعیت تأهل مالیات بر درآمد

	Refund	Marital Status	Taxable Income	Class
1	yes	Single	125k	No
2	No	Married	100k	No
3	No	Single	70k	No
4	yes	Married	120k	No
5	No	Single	95k	yes
6	No	Married	60k	No
7	yes	Single	220k	No
8	No	Single	85k	yes
9	No	Married	75k	No
10	No	Single	90k	yes

class →	No	No	No	yes	yes	yes	No	No	No	No
	60	70	75	85	90	95	100	120	125	220

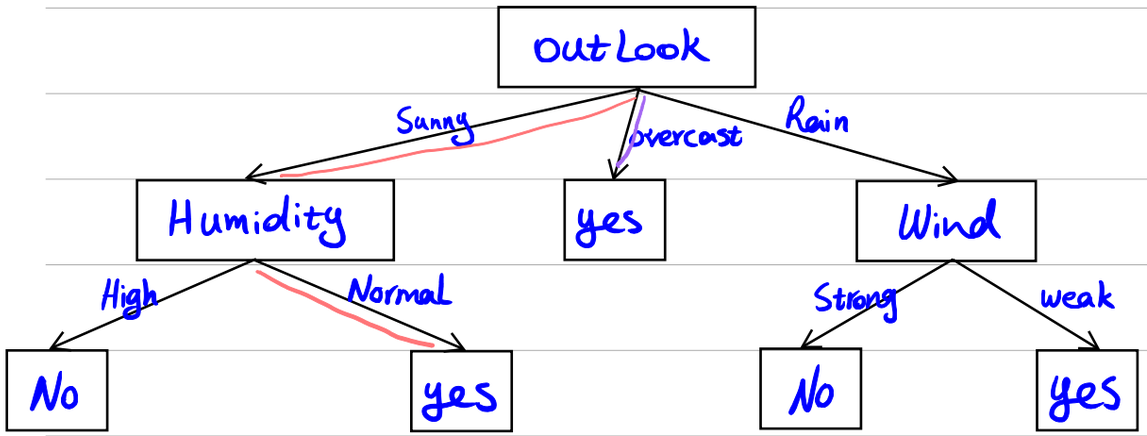
	55	65	72	80	87	92	97	110	122	172	230
	<= >										
yes	0 | 3	0 | 3	0 | 3	0 | 3	1 | 2	2 | 1	3 | 0	3 | 0	3 | 0	3 | 0	3 | 0
No	0 | 7	1 | 6	2 | 5	3 | 4	3 | 4	3 | 4	3 | 4	4 | 3	5 | 2	6 | 1	7 | 0

min

$$\text{GiniSplit}(97) = \underbrace{\frac{6}{10}}_{<=97}\left(1-\left(\frac{3}{6}\right)^2-\left(\frac{3}{6}\right)^2\right)+\underbrace{\frac{4}{10}}_{>97}\left(1-\left(\frac{0}{4}\right)^2-\left(\frac{4}{4}\right)^2\right)= 0.3$$

اما همانطور که احتمالاً حدس می زنید، دقت DT برای داده های پیوسته پایین است.

قوانین در DT:



به هر مسیر از ریشه به برگ می‌گویند یک قانون.

If (outLook = Sunny ∧ Humidity = Normal) Then Play = yes

If (outLook = overcast) Then Play = yes

؟: چه طور متوجه شویم که یک قانون درست است یا هم تر از بقیه است؟

	بازپرداخت وام قبلی	وضعیت تأهل	مالیات بر درآمد	می‌تواند وام بگیرد؟
	Refund	Marital Status	Taxable Income	Class
1	yes	Single	125k	No
2	No	Married	100k	No
3	No	Single	70k	No
4	yes	Married	120k	No
5	No	Single	95k	yes
6	No	Married	60k	No
7	yes	Single	220k	No
8	No	Single	85k	yes
9	No	Married	75k	No
10	No	Single	90k	yes

$$\text{Coverage}(R) = \frac{n_{covers}}{|D|}$$

$$\text{accuracy}(R) = \frac{n_{covers}}{n_{covers}}$$

If (Status = Single) Then class = No

$\text{Coverage} = 6/10$ ← چند نفر از کل مجرد هستند

$\text{accuracy} = 3/6 = 50\%$

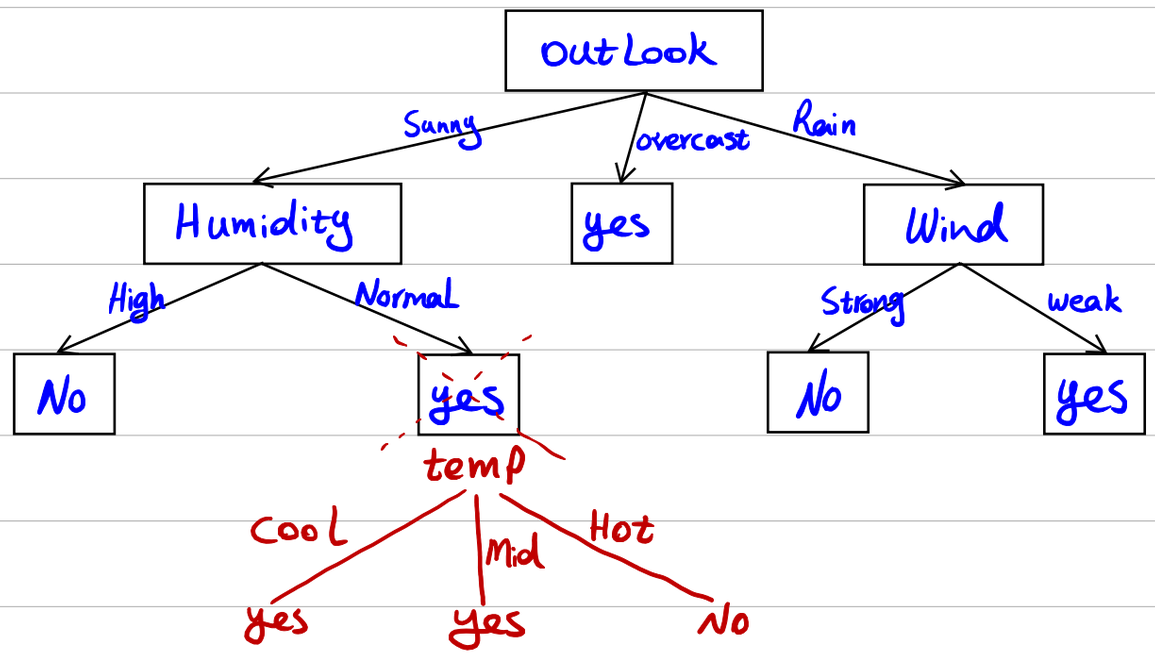
هرچه coverage و accuracy بیشتر باشند،

قانون ما باارزش‌تر است.

؟: DT در برابر noise چه طور رفتار می کند؟

outLook	Temp	Hummidity	Wind	Play=?
Sunny	Hot	Normal	Strong	No

→ نویز



اگر تعداد انشعاب ها درخت به صورت غیر قابل قبولی افزایش یابد، دچار overfitting خواهیم شد.

؟: چه طوری توانیم مشکل overfitting را برطرف کنیم؟

در درخت تصمیم (Decision Tree)، تنظیم **هایپرپارامترها** نقش مهمی در تعیین عملکرد و پیچیدگی مدل ایفا می‌کند. هایپرپارامترها مقادیر از پیش تعیین شده‌ای هستند که به مدل داده می‌شوند تا رفتار و ساختار درخت تصمیم را کنترل کنند. در ادامه مهم‌ترین هایپرپارامترهای درخت تصمیم را توضیح می‌دهم:

1. `max_depth` (عمق ماکسیمم درخت)

- **توضیح:** این پارامتر عمق حداکثری درخت تصمیم را مشخص می‌کند. عمق درخت به معنای تعداد لایه‌ها یا سطوحی است که گره‌های درخت می‌توانند داشته باشند.
- **تأثیر:** اگر مقدار `max_depth` کم باشد، مدل ساده‌تر می‌شود و احتمال **کم‌برازش (Underfitting)** وجود دارد. اگر این مقدار خیلی بزرگ باشد، مدل ممکن است بیش از حد پیچیده شود و به **بیش‌برازش (Overfitting)** دچار شود.

2. `min_samples_split` (حداقل تعداد نمونه‌ها برای تقسیم)

- **توضیح:** این پارامتر حداقل تعداد نمونه‌هایی که یک گره باید داشته باشد تا بتواند به دو گره تقسیم شود را مشخص می‌کند.
- **تأثیر:** اگر مقدار این پارامتر زیاد باشد، درخت کمتر تقسیم می‌شود و ساده‌تر باقی می‌ماند. این می‌تواند به جلوگیری از بیش‌برازش کمک کند. در مقابل، مقدار کم آن باعث می‌شود که گره‌ها به راحتی تقسیم شوند و مدل پیچیده‌تر شود.

3. `min_samples_leaf` (حداقل تعداد نمونه‌ها در هر برگ)

- **توضیح:** این پارامتر حداقل تعداد نمونه‌هایی که باید در یک گره برگ (Leaf Node) وجود داشته باشد را مشخص می‌کند.
- **تأثیر:** مقدار بزرگ‌تر این پارامتر باعث می‌شود که درخت تصمیم برگ‌های ساده‌تری بسازد، که منجر به جلوگیری از بیش‌برازش می‌شود. اگر مقدار آن کوچک باشد، برگ‌های بیشتری ساخته می‌شود و احتمال پیچیدگی بیشتر در مدل وجود دارد.

4. `max_features` (حداکثر تعداد ویژگی‌های مورد استفاده در هر تقسیم)

- **توضیح:** این پارامتر مشخص می‌کند که در هر تقسیم (Split) چه تعداد ویژگی برای انتخاب بهترین تقسیم در دسترس باشند. این ویژگی‌ها می‌توانند به صورت تصادفی از بین کل ویژگی‌ها انتخاب شوند.
- **تأثیر:** اگر این مقدار کمتر از تعداد کل ویژگی‌ها باشد، مدل تعمیم‌پذیری بهتری خواهد داشت، زیرا از **کاهش واریانس** (Variance Reduction) بهره‌مند می‌شود. در مقابل، اگر تعداد زیادی از ویژگی‌ها استفاده شوند، مدل ممکن است بیش‌برازش کند.

5. `max_leaf_nodes` (حداکثر تعداد برگ‌ها)

- **توضیح:** این پارامتر حداکثر تعداد گره‌های برگ را در درخت تصمیم محدود می‌کند.
- **تأثیر:** اگر این مقدار محدود باشد، مدل ساده‌تر و احتمال کم‌برازش بیشتر می‌شود. اما اگر مقدار آن زیاد باشد، مدل ممکن است بیش‌برازش کند و تعداد برگ‌ها افزایش یابد.

6. `min_weight_fraction_leaf` (حداقل وزن کسر نمونه‌ها در برگ)

- **توضیح:** این پارامتر مشخص می‌کند که حداقل چه کسر از وزن کل نمونه‌ها باید در هر گره برگ وجود داشته باشد.
- **تأثیر:** اگر نمونه‌های دارای وزن بالا باشند، این پارامتر می‌تواند از ایجاد برگ‌های کوچک جلوگیری کند و کمک به ساده‌تر کردن مدل نماید.

7. `criterion` (معیار انتخاب تقسیم)

- **توضیح:** این پارامتر معیار ارزیابی کیفیت تقسیم گره‌ها را تعیین می‌کند. دو معیار اصلی برای این کار وجود دارد:
- **Gini**: شاخص جینی برای اندازه‌گیری خلوص گره‌ها (برای داده‌های دسته‌بندی).
- **Entropy**: آنتروپی (یا بهره اطلاعات) برای محاسبه میزان نااطمینانی کاهش‌یافته پس از تقسیم (برای داده‌های دسته‌بندی).
- **تأثیر:** هر یک از این معیارها به روش‌های مختلفی کیفیت تقسیم گره‌ها را اندازه‌گیری می‌کنند و تأثیر مستقیمی بر عملکرد مدل دارند.

8. `splitter` (روش انتخاب تقسیم)

- **توضیح:** این پارامتر مشخص می‌کند که چگونه در هر گره ویژگی‌ها برای تقسیم انتخاب شوند. دو حالت اصلی وجود دارد:
- **Best**: بهترین تقسیم بر اساس معیار انتخاب می‌شود.
- **Random**: تقسیم به صورت تصادفی انتخاب می‌شود.
- **تأثیر:** انتخاب حالت **best** باعث بهبود عملکرد مدل می‌شود ولی ممکن است زمان بیشتری بگیرد، در حالی که **random** می‌تواند سرعت را بالا ببرد ولی کیفیت تقسیم ممکن است پایین‌تر باشد.

9. `ccp_alpha` (پیچیدگی هرس هزینه)

- **توضیح:** این پارامتر برای هرس درخت تصمیم استفاده می‌شود. هر چه این مقدار بزرگ‌تر باشد، درخت هرس بیشتری می‌شود و گره‌های کم‌اهمیت حذف می‌شوند.
- **تأثیر:** استفاده از این پارامتر می‌تواند پیچیدگی مدل را کاهش دهد و از بیش‌برازش جلوگیری کند.

10. `max_samples` (در نمونه‌برداری از داده‌ها)

- **توضیح:** این پارامتر در حالت‌های درخت تصمیم تصادفی استفاده می‌شود و مشخص می‌کند چه تعداد نمونه از داده‌ها به طور تصادفی برای آموزش مدل استفاده شوند.
- **تأثیر:** استفاده از این پارامتر به افزایش تعمیم‌پذیری مدل کمک می‌کند و باعث می‌شود مدل کمتر به داده‌های آموزشی وابسته باشد.

جمع‌بندی:

هایپرپارامترهای درخت تصمیم می‌توانند کنترل کنند که مدل چقدر پیچیده یا ساده باشد. تنظیم مناسب این پارامترها باعث می‌شود مدل بهتر عمل کند و از **بیش‌برازش** و **کم‌برازش** جلوگیری شود.

Pruning یا هرس کردن

Pre-Pruning یا پیش هرس

پیش‌هرس به این معنی است که فرآیند رشد درخت را قبل از اینکه درخت به صورت کامل ساخته شود، متوقف کنیم. این کار با تعیین محدودیت‌هایی انجام می‌شود که از ایجاد بیش از حد جزئیات در ساخت درخت جلوگیری می‌کنند

روش‌های رایج پیش‌هرس

Maximum Depth یا عمق ماکسیمم

محدود کردن عمق درخت به یک مقدار از پیش تعیین شده. در این حالت، درخت نمی‌تواند به تعداد بیش از حد زیادی از لایه‌ها گسترش یابد

Minimum Samples per Node یا حداقل تعداد نمونه‌ها در هر گره

گره‌های داخلی تنها زمانی شکافته می‌شوند که تعداد نمونه‌ها در هر گره بیشتر از یک مقدار مشخص باشد

Minimum Samples to Split یا حداقل تعداد نمونه‌ها برای تقسیم یک گره

اگر تعداد نمونه‌ها در یک گره کمتر از این مقدار باشد، آن گره دیگر تقسیم نمی‌شود و به عنوان برگ در نظر گرفته می‌شود

پیش‌هرس به جلوگیری از گسترش بیش از حد درخت کمک می‌کند، اما ممکن است منجر به ساده‌سازی بیش از حد مدل شود و برخی از الگوهای مهم را از دست دهد

Pruning یا هرس کردن

Post-Pruning یا پس هرس

پس‌هرس به این معنی است که ابتدا اجازه دهیم درخت به صورت کامل ساخته شود و سپس با اعمال تغییرات، بخش‌هایی از درخت را که اطلاعات کمتری دارند یا منجر به پیچیدگی اضافی می‌شوند، حذف کنیم. در این حالت، درخت ابتدا به طور کامل رشد می‌کند و سپس بخش‌های غیرضروری آن هرس می‌شوند

روش‌های رایج پس‌هرس

Pruning with Validation Set یا هرس با استفاده از اعتبارسنجی

در این روش، درخت را بر روی یک مجموعه داده اعتبارسنجی آزمایش می‌کنیم و شاخه‌هایی که منجر به بهبود قابل توجهی در عملکرد مدل نمی‌شوند، حذف می‌کنیم

Cost Complexity Pruning یا هرس با استفاده از هرس هزینه

این روش یک معیار پیچیدگی برای درخت تعیین می‌کند و شاخه‌هایی که به صورت جزئیات غیرضروری در درخت محسوب می‌شوند را حذف می‌کند، به طوری که مدل ساده‌تر شود بدون اینکه دقت آن کاهش یابد

پس‌هرس به ما اجازه می‌دهد که ابتدا درختی با جزئیات زیاد ایجاد کنیم و سپس آن را بهینه کنیم، که این به طور معمول دقت و تعمیم‌پذیری مدل را بهتر از پیش‌هرس می‌سازد

معیار	Pre-Pruning (پیش‌هرس)	Post-Pruning (پس‌هرس)
زمان اجرا	سریع‌تر (به دلیل توقف زودتر)	کندتر (ابتدا درخت کامل ساخته می‌شود)
پیچیدگی مدل	مدل ساده‌تر، ممکن است الگوهای پیچیده نادیده گرفته شوند	مدل ابتدا پیچیده و سپس بهینه‌سازی می‌شود
احتمال بیش‌برازش	کمتر	متوسط (پس از هرس بهبود می‌یابد)
کنترل فرآیند هرس	تعیین محدودیت‌های اولیه	هرس پس از ساخت کامل درخت
دقت نهایی مدل	ممکن است کمتر باشد	معمولاً بهتر بهینه می‌شود

مفهوم اصلی Random Forest:

رندوم فارست، همان‌طور که از نامش پیداست، از مجموعه‌ای از **درخت‌های تصمیم تصادفی** استفاده می‌کند. در این روش، به جای ساخت یک درخت تصمیم واحد، چندین درخت تصمیم ساخته می‌شوند و هر کدام به صورت مستقل از داده‌های مختلف (تحت فرایند نمونه‌برداری تصادفی) آموزش می‌بینند. سپس برای پیش‌بینی نهایی، از نتایج این درخت‌ها استفاده می‌شود.

5. هایپرپارامترهای مهم در Random Forest:

- `n_estimators` : تعداد درخت‌های تصمیم در جنگل. افزایش تعداد درخت‌ها دقت مدل را افزایش می‌دهد، اما ممکن است زمان محاسبه نیز بیشتر شود.
- `max_depth` : حداکثر عمق درخت‌ها. کنترل عمق درخت‌ها می‌تواند از بیش‌برازش جلوگیری کند.
- `min_samples_split` : حداقل تعداد نمونه‌هایی که برای تقسیم یک گره لازم است.
- `min_samples_leaf` : حداقل تعداد نمونه‌ها در یک برگ.
- `max_features` : تعداد ویژگی‌های تصادفی که در هر گره برای تقسیم در نظر گرفته می‌شوند. می‌تواند به عنوان یک نسبت از کل ویژگی‌ها یا یک عدد مشخص باشد.
- `bootstrap` : اگر مقدار آن `True` باشد، داده‌ها با جایگزینی نمونه‌برداری می‌شوند (بگینگ). اگر `False` باشد، نمونه‌برداری بدون جایگزینی است.

کاربردهای Random Forest:

- **دسته‌بندی**: به عنوان مثال در تشخیص بیماری، تشخیص اسپم، تشخیص تصویر و غیره.
- **رگرسیون**: در پیش‌بینی قیمت خانه‌ها، پیش‌بینی درآمد و سایر مسائل رگرسیونی.
- **تعیین اهمیت ویژگی‌ها (Feature Importance)**: این الگوریتم می‌تواند ویژگی‌هایی که بیشترین تأثیر را بر پیش‌بینی‌ها دارند شناسایی کند، که برای انتخاب ویژگی (Feature Selection) مفید است.