تمرین 5 یادگیری ماشین — علیرضا حبیب زاده 99109393

1.

۱_ اندازه‌ی مجموعه‌ی فرضیه بستگی به این دارد که چه نوع توابعی استفاده کنیم ولی در حالت کلی با فرض کلی بودن توابع مورد استفاده نهایتا 2^N حالت ممکن است به این دلیل که تعداد حالات خوب یا بد بودن عملکرد هر کدام از افراد است.

۲_

$$P[|E_{in}-E_{out}|<\epsilon] \leqslant M\, e^{-2\epsilon^2 N}$$

$$P[|E_{in}-E_{out}| \leqslant 0.02] \leqslant 2^{10000} \times e^{-2.0.02^2 \times 10000}$$

$$= e^{10000\ln 2 - 2\times 0.02^2 \times 10000}$$

$$= e^{10000(\ln 2 - 2\times 0.02^2)} > 1$$

پس چیز مفیدی نمی‌گوید. تا وقتی M را محدود نکرده‌ایم چیز خاصی نمی‌گوید.

۱_ تنها یکبار داده‌ها را sample گرفتیم و فرضیه از یک سری فرمول های ریاضی بدست آمده نه یادگیری پس مجموعه فرضیه(ها) تنها یک عضو دارد: $\boxed{M=1}$

۲_ با این تفاسیر باند هافدینگ برای N=1000 چنین می‌شود:

$$P[|\cancelto{0}{E_{in}}-E_{out}|<\epsilon] \leqslant 2\times(1)\times e^{-\epsilon^2 N}$$

$$P\left[\,\overbrace{|\underbrace{E_{in}}_{0} - E_{out}|}^{=E_{out}} < 0.02\right] \leqslant 2e^{-0.02^2 \times 10000}$$

$$= 0.0366 = 3.66\%$$

می‌گوید احتمال این اتفاق (بخش 2) $\boxed{\text{بیشتر مساوی } 96.34\%}$ است.

3 - اولاً که اگر برای 100 بانک این کار را انجام دهیم برای حدود 4 تای آن‌ها به طور متوسط خطا بیشتر از 2% خواهد بود البته گفته شده بیش از نیمی به تأخیر افتادند پس خیلی خیلی نامحتمل است و مشکل از جای دیگر است.

اولین دلیل ممکن به نظرم bias بودن داده‌ی ورودی است. ما تنها آن‌هایی را به عنوان ورودی به تابع دادیم و تست کردیم که توسط بانک بیشتر تأیید شده بودند. پس الگوریتم (توابع ما) تا به حال عملکرد پس از دادن کریدیت به آن‌هایی را که توسط سیستم قبلی بانک رد می‌شدند ندیده. احتمالاً بعد از (پس از به کارگیری مدل توسط بانک) ورودی‌ای داده می‌شود که بیشتر توسط بانک رد می‌شوند و هیچ اثری از آن‌ها در دیتاست ورودی نیست. به نظرم این‌ها هستند که تأثیر زیادی روی خطای نهایی گذاشته‌اند.

احتمال دیگر این است که مدل به دلیل پیچیدگی زیاد به داده‌ها overfit شده اما

به نظرم این خیلی نامحتمل است چرا که تعداد داده ها زیاد بوده و اصلا یادگیری رخ نداده و ما بدون نگاه کردن به داده ها یک تابع ساختیم و آن را روی داده‌ها اعمال کردیم و $E_{in} = 0$ گرفتیم. و اصلا با بدها فدینگ قرار است همین امکان اورفیت را پیش‌بینی کند ($|E_{out} - E_{in}|$) که می‌گوید نامحتمل است پس مشکل همان بایاس بودن داده ها بوده.

4 - به دلیل این که می‌توان روش اولیه (legacy) بانک را نگه داشت و از آن در کنار این مدل استفاده کرد. آن هایی که در روش اولیه تایید می‌شوند حال توسط تابع ما ارزیابی می‌شوند و مدل ما با دقت خیلی بالایی (حال تقریبا همان باند هافدینگ بالا) عملکرد این ها را پیش‌بینی می‌کند و حال می‌توان آن هایی که در این رد می‌شوند رد کرد.

با این کار برای آن درصدی که در روش اولیه تایید می‌شوند خطای ما بهتر می‌شود و اطمینان احتمالاتی هم داریم. برای آن هایی که در روش اولیه رد می‌شوند هم خطا همان خطای قبلی است.

آن هایی که درست اولیه رد می‌شوند را یا می‌توان رد کرد (تا تعداد رد شده ها کم زیاد شود) یا می‌توان درصدی از آن ها که تابع ما در مورد آن ها خیلی مطمئن است را تایید کرد. این جوری حتی اگر درصدی از آن ها fail شوند درصد محدودی

هستند و اطمینان احتمالاتی داریم نسبت به درصدی از آن‌ها که تایید می‌کنیم.

البته این به شرطی بود که اصرار به استفاده از همان تابع داریم. اگر می‌توانیم تابع را عوض کنیم می‌توانیم تعدادی از رد شده‌ها را هم کریت بدهیم و عملکرد آن‌ها را بررسی کنیم. ممکن است بگیم $E_{in} = 0$ نشود اما باز هم فیتینگ درستی داریم.

یا می‌توان تعداد زیادتری تایید شده‌ی اولیه را جوری سمپل کرد که توزیع متغیرهای آن‌ها مثل سن، درآمد و ... ، x ها، مانند توزیع همه‌ی افراد (از جمله تایید اولیه نشده‌ها) شود. این گونه bias را از بین می‌بریم.

2. ۱ـ m ویژگی که هرکدام می‌توانند دو مقدار اختیار کنند 2^m حالت ممکن

برای ویژگی‌ها به وجود می‌آورد. با استفاده از درخت می‌توان هرکدام از آن‌ها را از دیگری

تمییز داد و 2^m برگ در درخت داشت. هرکدام از 2^m برگ درخت را می‌توان

در یکی از دو دسته‌ی مسئله‌ی دسته‌بندی قرار داد و هرکدام یک درخت جدید می‌دهد پس

2^{2^m} درخت متفاوت داریم.

۲ـ خیر تنها درصورتی که همه‌ی اشکال مختلف درخت‌ها را در نظر بگیریم و برای

هرکدام بهترین برگ‌ها را قرار دهیم و ببینیم کدام درخت بهتر بوده می‌توانیم

بهترین درخت را بیابیم (که غیرممکن است)

کارهای حریصانه‌ای که local انجام شوند تنها ما را به بهینه‌ی local می‌رسانند

و ممکن است ما را از بهینه‌ی global دور کنند. همچنین درخت‌های تصمیم

ممکن است به نویز و داده‌های پرت حساس باشند. ممکن است تصمیم حریصانه‌ی

ما که در لحظه Information Gain مثبتی دارد ناشی از یک داده‌ی پرت باشد که نهایتا

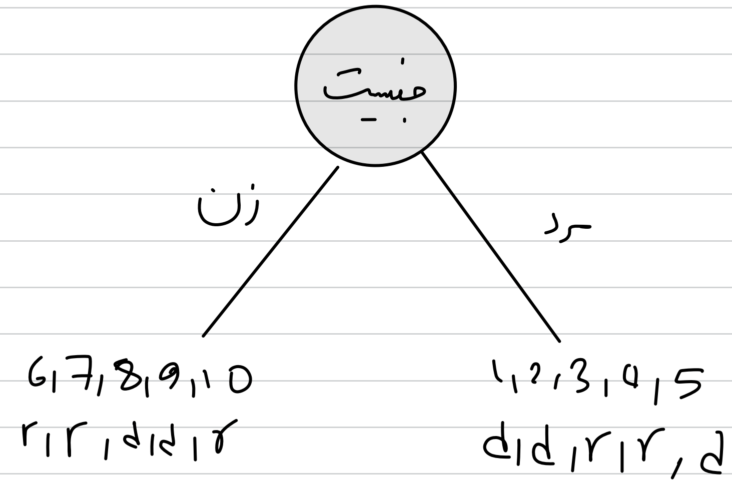
با دیدن همه‌ی داده‌ها تصمیم بدی باشد.

۳ـ آ) ابتدا نگاه می‌کنیم در مرحله‌ی اول درخت تصمیم کدام یک از ویژگی‌های

جنسیت، نژاد یا محل زندگی را قرار دهیم. Gini Impurity کمتری دارد.

d = دموکرات

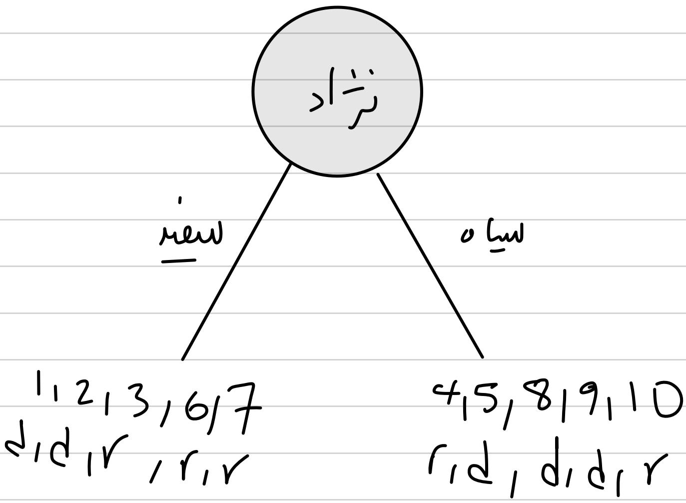
r = جمهوری‌خواه



$$gini(\text{زن}) = 1-\left(\frac{3}{5}\right)^2-\left(\frac{2}{5}\right)^2 = \frac{12}{25}$$

$$gini(\text{مرد}) = 1-\left(\frac{2}{5}\right)^2-\left(\frac{3}{5}\right)^2 = \frac{12}{25}$$

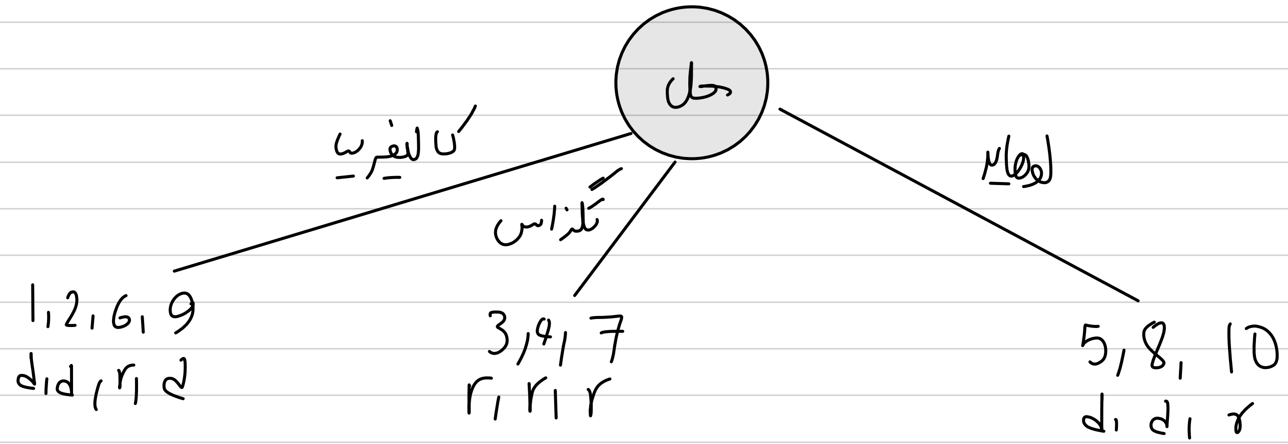
$$gini = \frac{5}{10}\left(\frac{12}{25}\right)+\frac{5}{10}\left(\frac{12}{25}\right) = \frac{12}{25} = \boxed{0.48}$$



$$gini = \boxed{0.48}$$

ترکیب مشابه قبل است پس:

شکل



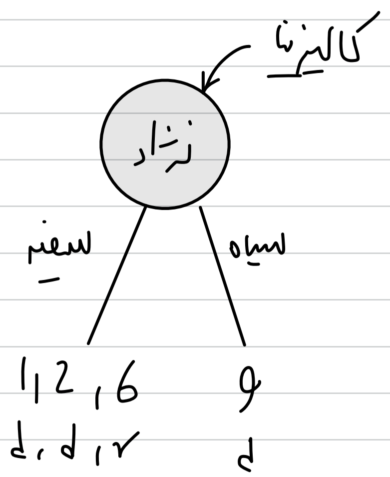
$$gini(\text{اوهایو}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$gini(\text{تگزاس}) = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$gini(\text{کالیفرنیا}) = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = \frac{3}{8}$$

$$gini = \frac{4}{9} \times \frac{3}{10} + \frac{3}{8} \times \frac{4}{10} = \frac{17}{60} = \boxed{0.28\overline{3}}$$

پس محل زندگی بهترین تقسیم اول است. حال برای تقسیم دوم هر کدام:



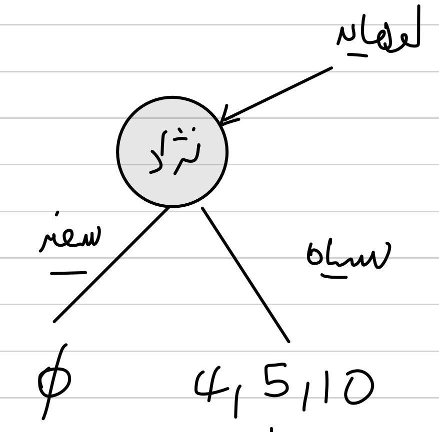
$$1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 4/9 \qquad 0$$

$$4/9 \times \frac{3}{4} = 1/3$$

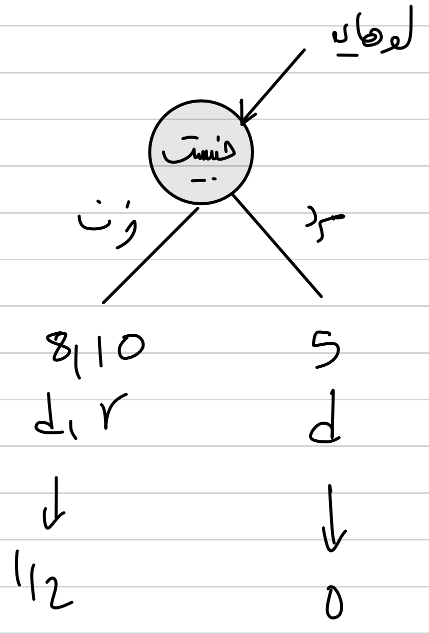
$$1 - \left(\frac{1}{2}\right)^2 - (1/2)^2 = 1/2 \qquad 0$$

$$\frac{1}{2} \times \frac{2}{4} = \frac{1}{4}$$ ✓

گلزاس نصف قبل ۵ شد پس فرقی نمی‌کند چه انتخاب کنیم (یا اصلا همینجا ببریم و ۲ اعلام می‌کنیم.

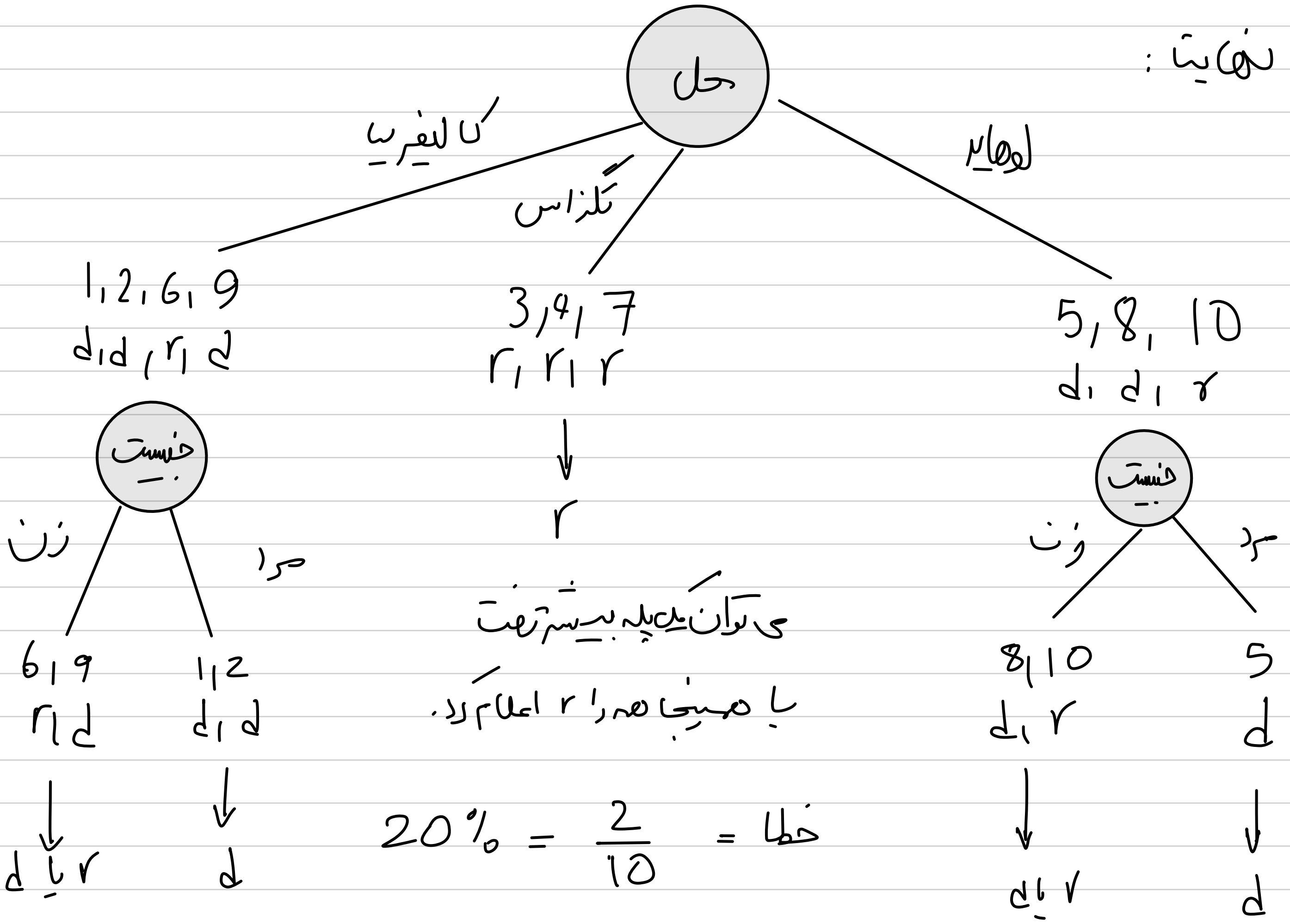


$= 1 - (1/3)^2 - (2/3)^2 = 4/9$



$1/2 \times \frac{2}{3} = 1/3$ ✓

نهایتا:

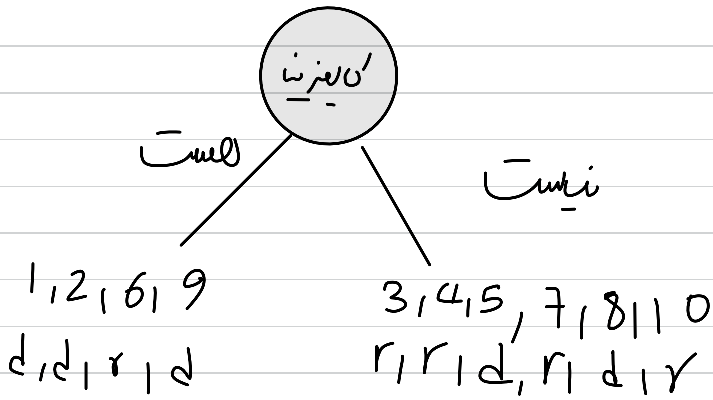


می‌توان یک پله بیشتر رفت

یا همینجا همه را r اعلام کرد.

$20\% = \frac{2}{10} =$ خطا

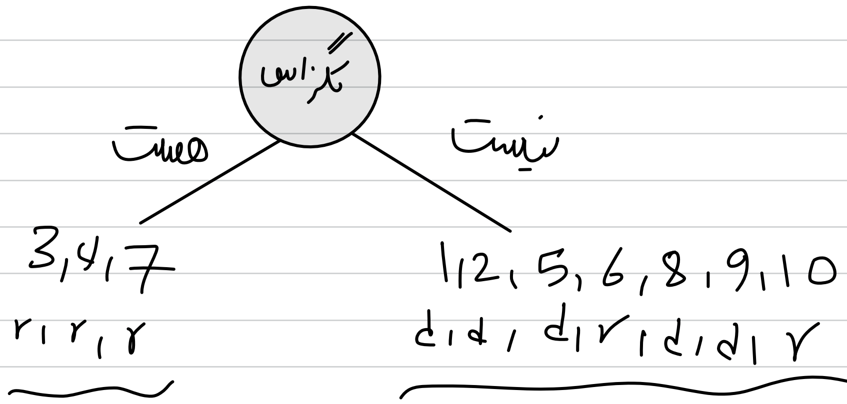
ب) معیارهای قبلی برای برش اول همه binary بودند جز محل زندگی که حالات آن را بررسی می‌کنیم. چون تنها سه مقدار ممکن داریم تمام ترکیب شرطی منطقی بودن یا قبول هر مکان است. (اگر تگزاس هست یا نه) و حالات دیگر (مثل کالیفرنیا یا اوهایو هست یا نه) به همان حالات ساده ترجمه می‌شوند.



$$gini(\text{هست}) = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = \frac{3}{8}$$

$$gini(\text{نیست}) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{4}{9}$$

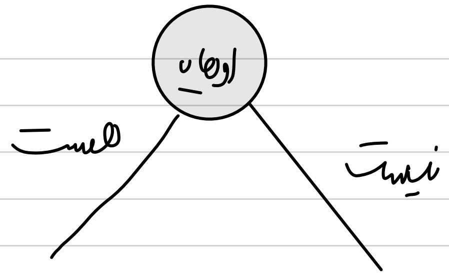
$$\frac{3}{8} \times \frac{4}{10} + \frac{4}{9} \times \frac{6}{10} = \frac{5}{12} \simeq 0.417$$



$$0$$

$$1 - \left(\frac{5}{7}\right)^2 + \left(\frac{2}{7}\right)^2 = \frac{20}{49}$$

$$gini = \frac{20}{49} \times \frac{7}{10} = \frac{2}{7} \simeq 0.286$$



5, 8, 10
d, d, r

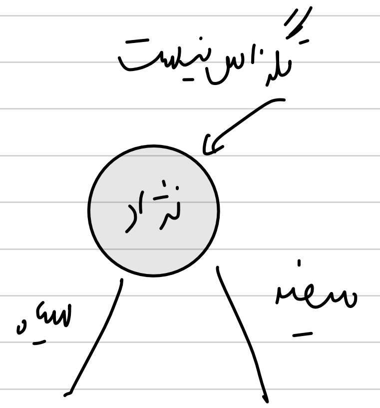
$\frac{4}{9}$

1, 2, 3, 4, 6, 7, 9
d, d, r, r, r, r, d

$1-\left(\frac{3}{7}\right)^2-\left(\frac{4}{7}\right)^2=\frac{24}{49}$

$gini=\frac{4}{9}\times\frac{3}{10}+\frac{24}{49}\times\frac{7}{10}=\frac{10}{21}\simeq 0.476$

پس تصمیم اول تگزاس هست یا نه است. (دو تا ی نژاد و جنسیت را امتحان کردیم
دیدیم و جینی 0.48 داشتند) برای تصمیم دوم بخش تگزاس هست:
سمت تگزاس هست را با محل زندگی نمی‌گیریم ! چون تگزاس هست
دیگه اضافه کردن مثلا کالیفرنیا هست هم داره نیست میبرد و از آنها بهتر نمی‌شود.
اصلا تگزاس هست همه r هستند. برای نیست:



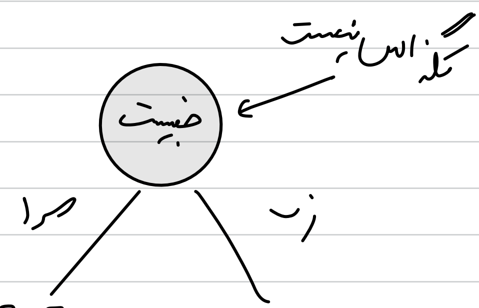
5, 8, 9, 10
d, d, d, r

$2\times\frac{3}{4}\times\frac{1}{4}=3/8$

1, 2, 6
d, d, r

$\frac{4}{9}$

$3/8\times\frac{4}{7}+\frac{4}{9}\times\frac{3}{7}=\frac{17}{42}\simeq 0.405$



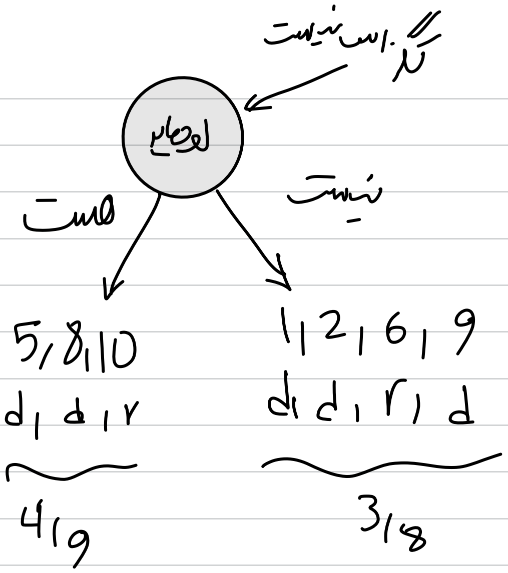
1, 2, 5
d, d, d

0

6, 8, 9, 10
r, d, d, r

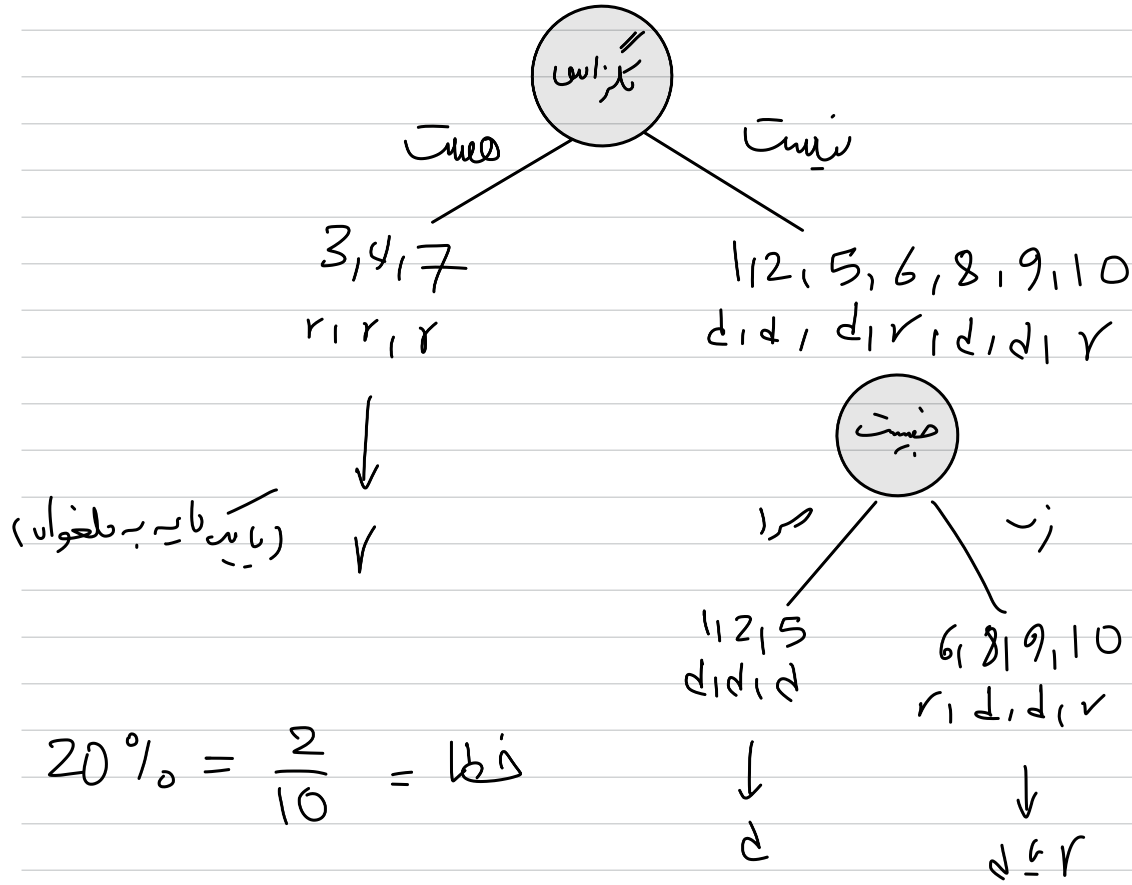
1/2

$1/2\times\frac{4}{7}=2/7=0.286$



$$\Rightarrow \frac{17}{42} \approx 0.405$$

پس نهایتا :



$$20\% = \frac{2}{10} = \text{خطا}$$

ج) به نظرم یکسان شدن خطای این دو اینجا تصادفی و از روی خوش شانسی

استراتژی binary است. معمولا این استراتژی multiway به عمق کمتری

می‌انجامد و یعنی در عمق یکسان خطای کمتری دارد. برخی مزایای multiway :

— تقسیم طبیعی : بعضی ها مثل محل زندگی دارای جوابی به طور طبیعی

چون حالت های بیشتر در multiway نوعی تقسیم طبیعی و مشهودی تر است.

- احتمالا درخت ها عمق کمتری خواهند داشت چون قدرت multiway بیشتر است.

معایب multiway:

- ریسک over fit شدن در آن بیشتر است چون پیچیدگی مدل بیشتر است.

- پیاده سازی سخت تری دارد هم در پیاده سازی و هم در اجرا چون چک کردن یک شرط هم ساده که و هم از نظر سخت افزاری سریع تر است.

مزایای binary:

- در کل می دانیم مدل ساده تری که کار کند بهتر است. اگر binary کار کند خب مدل ساده تر استفاده کنیم.

- در حالت کلی multiway را هم شامل می شود. اگر اجازه دهیم در عمق $\log C$ که C تعداد طبقه بندی ها است همان کار multiway را می کند. اما عیب این که عمق زیاد شده و مزیت این که هر شاخه لزومی ندارد مثل قبل پیچیده باشد و می توان در مثل دیگر لازم نیست همان ابتدا به ۲ برود.

- می توان آن را به صورت عدد باینری encode کرد که در پردازش خیلی

خوب است.

_ اینجا overfit غلط است چون الگوریتم بعد VC کمتری دارد و ساده‌تر است.

مطلب هم در متن مزایای قبلی گفتم. ممکن است برخی تقسیم های طبیعی را از دست بدهد، شهودی نباشد یا عمق درخت زیاد شود.

4 _ الگوریتم جنگل تصادفی یک آنسامبل از درخت هاست مانند و بنابراین کار واریانس را کاهش میدهد. اینجا یک دسته از درخت ها واریانس کمتری نسبت به آموزش یک تک درخت دارند. چرا که برای مثال اگر یک کمیت را میانگین بگیریم:

$$Var\left(\frac{\sum x_i}{N}\right) = \frac{1}{N^2} Var\left(\sum x_i\right) = \frac{1}{N} Var(x_1)$$

و در حالت کلی و پیچیده‌تر هم این به طور تقریبی برقرار است.

در کنار این درخت های مختلف که هم روی فیچرهایی که میبینند تصادفی شده اند و هم روی قسمت هایی از داده که میبینند امکان کمتری دارد درخت نهایی روی نویز یاد داده شود و فیت به نویز نمی‌شود. مقدار زیادی از زیاد بودن واریانس درخت تکی فیت شدن به نویز است که هم با bagging (نگه داشتن بهترین بین چند درخت) و هم میانگین گیری و هم تصادفی سازی این کاهش می‌یابد. ضمنا احتمال overfit هم کم می‌شود.

3. ۱- نشان می‌دهیم به ازای هر i نامساوی برای جمله‌های متناظر برقرار است.

$$\mathbb{I}(\hat{y}_i \neq y_i) = \begin{cases} 0 & \hat{y}_i = y_i \quad (\text{sign}(H_t(x_i)) = y_i) \\ 1 & \hat{y}_i \neq y_i \quad (\text{sign}(H_t(x_i)) \neq y_i) \end{cases}$$

$$e^{-y_i H_t(x_i)} = \begin{cases} e^{-|H_t(x_i)|} & \hat{y}_i = y_i \\ e^{+|H_t(x_i)|} & \hat{y}_i \neq y_i \end{cases}$$

$$\begin{cases} > 0 \\ > 1 \end{cases}$$

پس در هر دو صورت: $e^{-y_i H_t(x_i)} > \mathbb{I}(\hat{y}_i \neq y_i)$

$$\Longrightarrow \frac{1}{N}\sum_i \exp(-y_i H_t(x_i)) \geq \frac{1}{N}\sum_i \mathbb{I}(\hat{y}_i \neq y_i)$$

2 ـ هرکدام از D_i ها توسط عامل‌های اسلید ضرب می‌شوند (بسته به درست یا غلط بودن پیش‌بینی بزرگ یا کوچک می‌شوند) با ضرب عامل y_i خودکار انجام می‌شود) :

و نهایتاً با Z نرمال می‌شود :

$$D_{t+1}(i) = \frac{D_t(i)\, e^{-\alpha_t y_i h_t(x_i)}}{\sum_i D_t(i) e^{-\alpha_t y_i h_t(x_i)}} = Z_t^{-1} D_t(i) e^{-\alpha_t y_i h_t(x_i)}$$

$$H_t(x_i) = \sum_{t=1}^{T} \alpha_t h_t(x_i)$$ 3-

با جایگذاری این عبارت در E و جابجا کردن ثوابت با جمع:

$$E = \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i H_t(x_i)) = \sum_{i=1}^{N} \frac{1}{N} \exp\left[\sum_{t=1}^{N} -\alpha_t y_i h_t(x_i)\right]$$

ضرب در جمع

ضرب در جمع

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T-1} Z_t \times \sum_{i=1}^{N} D_T(i)\, e^{-\alpha_T y_i h_T(x_i)}$$ 4-

حال $D_T(i)$ را از چیزی که در 2 بدست آمد جایگذاری می‌کنیم:

$$D_T(i) = Z_{T-1}^{-1} D_{T-1}(i)\, e^{-\alpha_{T-1} y_i h_{T-1}(x_i)}$$

$$\Longrightarrow \prod_{t=1}^{T} Z_i = \prod_{t=1}^{T-1} Z_t \cdot \sum_{i=1}^{N} Z_{T-1}^{-1} D_{T-1}(i)\, e^{-\sum_{t=T-1}^{T} \alpha_t y_i h_t(x_i)}$$

با عبارت آخر خط می‌خورد

$$= \prod_{t=1}^{T-2} Z_t \cdot \sum_{i=1}^{N} D_{T-1}(i)\, e^{-\sum_{t=T-1}^{T} \alpha_t y_i h_t(x_i)}$$

حال دوباره $D_{T-1}(i)$ را بر حسب قبلی می‌نویسیم:

$$D_{T-1}(x_i) = Z_{T-2}^{-1} D_{T-2}(x_i)\, e^{-\alpha_{T-2} y_i h_{T-2}(x_i)}$$

پس با استقرا اگر همین طور عقب رفت تا :

$$\prod_{t=1}^{T} Z_t = \sum_{i=1}^{N} \cancelto{1/N}{D_1(i)} \exp\left(\sum_{t=1}^{T} -\alpha_t y_i h_t(x_i)\right)$$

اما در ابتدا داده‌ها یکسان وزن داده شده اند و $D_1(i) = \frac{1}{N}$ پس :

$$\prod_{t=1}^{T} Z_t = \sum_{i=1}^{N} \frac{1}{N} \exp\left(\sum_{t=1}^{T} -\alpha_t y_i h_t(x_i)\right) = E$$

که همان عبارت بخش 3 است .

---

5 ـ توزیع وزن مرحله t هم نرمالایز است پس :

$$1 - \epsilon_t = \Big[\overbrace{\sum_i D_t(i)}^{1}\Big] - \Big[\sum_i D_t(i)\, \mathbb{I}(h_t(x_i) \neq y_i)\Big]$$

$$= \sum_i D_t(i)\left[1 - \mathbb{I}(h_t(x_i) \neq y_i)\right]$$

$$= \sum_i D_t(i)\, \mathbb{I}(h_t(x_i) = y_i)$$

$$\Longrightarrow (1-\epsilon_t)\, e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$$

$$= \sum_i D_t(i)\, \mathbb{I}(h_t(x_i) = y_i)\, e^{-\alpha_t} + \sum D_t(i)\, \mathbb{I}(h_t(x_i) \neq y_i)\, e^{+\alpha_t}$$

$$= \sum_i D_t(i)\Big[\underbrace{\mathbb{I}(h_t(x_i)=y_i)\,e^{-\alpha_t} + \mathbb{I}(h_t(x_i)\neq y_i)\,e^{+\alpha_t}}_{=\, e^{-\alpha_t y_i h_t}}\Big]$$

چون هرکدام از دسته‌بندهای ضعیف ۱+ یا ۱- خروجی می‌دهند و ضرب آن در y_i درصورت درست بودن ۱+ و درصورت غلط بودن ۱- است.

$$\Rightarrow (1-\epsilon_t)\,e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = \sum D_t(i)\,e^{-\alpha_t y_i h_t} = Z_t$$

۶_ درخصوص قبل دیدیم $Z_t = (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{+\alpha_t}$

در حد $\alpha_t \to \pm\infty$ ، $Z_t \to +\infty$ چون هربار یکی از نمایی‌ها از بین رفته و دیگری به $+\infty$ می‌رود و ϵ_t و $1-\epsilon_t$ مثبت است پس کمینه‌ی Z_t "حتماً" یکی از نقاط موضعی‌اش است:

$$\frac{\partial Z_t}{\partial \alpha_t} = -(1-\epsilon_t)\,e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = 0$$

$$(1-\epsilon_t) = \epsilon_t e^{2\alpha_t}$$

$$\Rightarrow \boxed{\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}} = \ln\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$$

چون در مرزها $(\pm\infty)$ تابع به $+\infty$ رفته تنها نقطه‌ی موضعی آن همان کمینه‌ی گلوبال است. (چون نرم و پیوسته و ـ ـ ـ ـ هم هست)