

به نام حضرت دوست



دانشگاه امیرکبیر  
دانشکده مهندسی کامپیوتر

---

داده کاوی

تمرین سری سوم

---

محمدحسین بدیعی

شماره دانشجویی 9531701

گرایش : هوش مصنوعی و رباتیک

استاد : دکتر احسان ناظر فرد

بهار 1400

## پاسخ سوال 1

(الف)

در واقع این احتمال به نظرم می‌تواند یک احتمال منطقی و معقول باشد زیرا باید توجه داشت که در دسته بندی ایمیل‌ها، اینکه یک ایمیل spam, regular طبقه بندی شود خیلی بدتر است از آنکه مثلاً یک ایمیل spam به عنوان regular دسته بندی شود و این مطلب کاملاً واضح است. لذا از نظر من این احتمال یک احتمال منطقی است.

(ب)

'study'	'free'	'money'	Category
1	0	0	Regular
0	0	1	Regular
1	0	0	Regular
1	1	0	Regular
0	1	0	Spam
0	1	0	Spam
1	1	0	Spam
0	1	0	Spam
0	1	1	Spam
0	1	1	Spam
0	1	1	Spam
0	0	1	Spam

ابتدا احتمالات را محاسبه می‌کنیم.

$$P(study|spam) = \frac{1}{8}$$

$$P(study|regular) = \frac{3}{4}$$

$$P(free|spam) = \frac{7}{8}$$

$$P(free|regular) = \frac{1}{4}$$

$$P(money|spam) = \frac{1}{2}$$

$$P(money|regular) = \frac{1}{4}$$

حال بر اساس قاعده بیز و فرضیات موجود در سوال داریم. (دقت شود از مخرج فورمول موجود در این قاعده طبق اسلایدهای درس بدلیل یکسان بودن در تعلق به مجموعه‌ی اسپم و غیر اسپم صرف نظر می‌کنیم چون برای محاسبه‌ی تعلق به اسپم و غیر اسپم، مخرج یکسان است.

$$P(s|spam) * P(spam) = P(study|spam) * P(1 - P(free|spam)) * P(money|spam) * P(spam)$$

$$\Rightarrow P(s|spam) * P(spam) = \frac{1}{8} * \left(1 - \frac{7}{8}\right) * \frac{1}{2} * \frac{1}{10} = 0.00078125$$

مجددا دقت کنید که ما صورت‌ها را محاسبه می‌کنیم (چون مخرج‌های یکسان است و حکم نرمالیزیشن را دارد)

حال برای regular بودن جمله نیز داریم.

$$P(s|regular) * P(regular) = P(study|regular) * P(1 - P(free|regular)) * P(money|regular) * P(regular)$$

$$\Rightarrow P(s|regular) * P(regular) = \frac{3}{4} * \frac{1}{4} * \frac{1}{4} * \frac{9}{10} = 0.0421875$$

لذا همانطور که می‌بینید، احتمال آنکه جمله‌ی مذکور در regular قرار گیرد بیشتر خواهد بود و لذا در دسته بندی جز regular قرار خواهد گرفت.

توجه کنید که برای بدست آوردن احتمال تعلق به مجموعه بر اساس قانون بیز باید اعداد بدست آمده‌ی فوق را نرمالیزه کنیم که تبدیل به احتمال شوند ولی چون مخرج در این قانون یکی است، لذا آن را به عنوان ترم نرمالیزه در نظر گرفتیم و صورت‌ها را محاسبه کردیم ولی اگر بخواهیم احتمالات را بیابیم داریم:

$$P(spam|s) = \frac{P(s|spam) * P(spam)}{P(s|spam) * P(spam) + P(s|regular) * P(regular)} = 0.018$$

$$P(regular|s) = \frac{P(s|regular) * P(regular)}{P(s|spam) * P(spam) + P(s|regular) * P(regular)} = 0.982$$

لذا این جمله متعلق در دسته‌ی Regular قرار خواهد گرفت.

## پاسخ سوال 2

برای حل این بخش می‌بایست information gain را برای هر یک از نقاط بدست آورده و آنکه مقدارش بالاتر بود انتخاب شود.

در ابتدا آنتروپی را برای D حساب می‌کنیم.

$$Entropy(D) = - \sum_{i=1}^n p_i \log_2 p_i = - \frac{9}{16} * \log_2 \frac{9}{16} - \frac{7}{16} * \log_2 \frac{7}{16} = 0.9887$$

سپس information gain را برای حالتی که نقطه‌ی اول را انتخاب کنیم محاسبه می‌کنیم.

$$E(value_{data} < point_1) = - \sum_{i=1}^n p_i \log_2 p_i = 0$$

$$E(value_{data} > point_1) = - \sum_{i=1}^n p_i \log_2 p_i = - \frac{7}{14} * \log_2 \frac{7}{14} - \frac{7}{14} * \log_2 \frac{7}{14} = 1$$

$$Gain(D, C_1) = Entropy(D) - \sum_v \frac{|D_v|}{|D|} Entropy(D_v) = 0.9887 - \frac{2}{16} * 0 - \frac{14}{16} * 1 = 0.1137$$

حال information gain را برای حالتی که نقطه‌ی دوم را انتخاب کنیم محاسبه می‌کنیم.

$$E(value_{data} < point_2) = - \sum_{i=1}^n p_i \log_2 p_i = - \frac{6}{9} * \log_2 \frac{6}{9} - \frac{3}{9} * \log_2 \frac{3}{9} = 0.9183$$

$$E(value_{data} > point_2) = - \sum_{i=1}^n p_i \log_2 p_i = - \frac{6}{7} * \log_2 \frac{6}{7} - \frac{1}{7} * \log_2 \frac{1}{7} = 0.5917$$

$$Gain(D, C_2) = Entropy(D) - \sum_v \frac{|D_v|}{|D|} Entropy(D_v) = 0.9887 - \frac{9}{16} * 0.9183 - \frac{7}{16} * 0.5917 = 0.2133$$

همانطور که مشاهده می‌کنید information gain به ازای نقطه‌ی دوم بیشتر از نقطه‌ی اول است و لذا نقطه‌ی دوم برای نقطه‌ی شکست مناسب تر است.

### پاسخ سوال 3

در ابتدا آنتروپی را برای D حساب می‌کنیم.

$$Entropy(D) = - \sum_{i=1}^n p_i \log_2 p_i = - \frac{10}{15} * \log_2 \frac{10}{15} - \frac{5}{15} * \log_2 \frac{5}{15} = 0.9183$$

در این مرحله 4 attribute داریم.

★ Attribute : زمان

$$D_{صبح} = [2F, 0N] \rightarrow E = 0$$

$$D_{بعد از ظهر} = [6F, 4N] \rightarrow E = - \frac{6}{10} \log_2 \frac{6}{10} - \frac{4}{10} \log_2 \frac{4}{10} = 0.971$$

$$D_{شب} = [2F, 1N] \rightarrow E = - \frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.9183$$

$$\Rightarrow Gain(D, \text{زمان}) = 0.9183 - \frac{10}{15} \times 0.971 - \frac{3}{15} \times 0.9183$$

$$\Rightarrow Gain(D, \text{زمان}) = 0.087$$

★ Attribute : نوع مسابقه

$$D_{مستد} = [3F, 3N] \rightarrow E = 1$$

$$D_{مکرر و آهسته} = [6F, 1N] \rightarrow E = - \frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = 0.5917$$

$$D_{دوستانه} = [1F, 1N] \rightarrow E = 1$$

$$\Rightarrow Gain(D, \text{نوع مسابقه}) = 0.9183 - \frac{6}{15} \times 1 - \frac{7}{15} \times 0.5917 - \frac{2}{15} \times 1 = 0.109$$

$$\Rightarrow Gain(D, \text{نوع مسابقه}) = 0.109$$



★ Attribute : زیست سابقه

$$D_{\text{رجن}} = [4F, 0N] \rightarrow E = 0$$

$$D_{\text{شی}} = [2F, 3N] \rightarrow E = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$D_{\text{ترکبی}} = [0F, 2N] \rightarrow E = 0$$

$$D_{\text{بند}} = [4F, 0N] \rightarrow E = 0$$

$$\Rightarrow \text{Gain}(D, \text{زیست سابقه}) = 0.9183 - \frac{5}{15} \times 0.971 = 0.5946$$

★ Attribute : حلاله قدرت

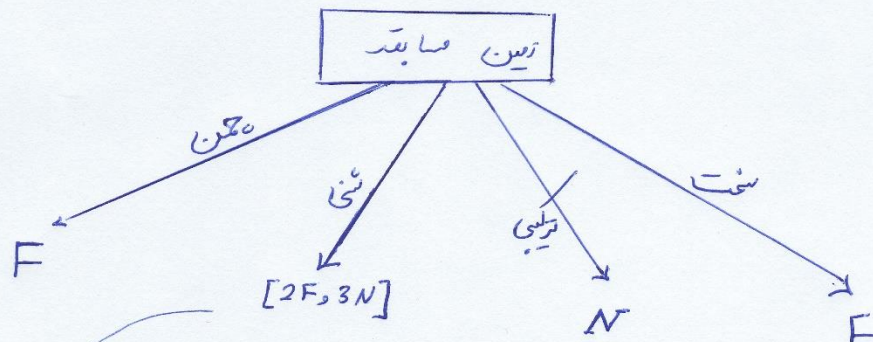
$$D_1 = [9F, 4N] \Rightarrow E = -\frac{9}{13} \log_2 \frac{9}{13} - \frac{4}{13} \log_2 \frac{4}{13} = 0.8905$$

$$D_2 = [1F, 1N] \Rightarrow E = 1$$

$$\Rightarrow \text{Gain}(D, \text{حلاله قدرت}) = 0.9183 - \frac{13}{15} \times 0.8905 - \frac{2}{15} \times 1$$

$$\Rightarrow \text{Gain}(D, \text{حلاله قدرت}) = 0.0132$$

$$\Rightarrow \left\{ \begin{array}{l} \text{Gain}(D, \text{زبان}) = 0.087 \\ \text{Gain}(D, \text{نوع سابقه}) = 0.109 \\ \text{Gain}(D, \text{زیست سابقه}) = 0.5946 \checkmark \rightarrow \text{Best attribute for root} \\ \text{Gain}(D, \text{حلاله قدرت}) = 0.0132 \end{array} \right.$$



$$Entropy(D) = 0.971$$

$\downarrow$   
 $D_{new}$

حال ادامه را در صورتیکه رین مسابقت ثنی باشد ادامه می دهیم.

★ Attribute زمان

$$D_{صبح} = [0F, 0N] \rightarrow E = 0$$

$$D_{بعد از ظهر} = [2F, 3N] \rightarrow E = 0.971$$

$$D_{شب} = [0N, 0F] \rightarrow E = 0$$

$$\Rightarrow Gain(D, \text{زمان}) = 0$$

$\downarrow$   
 $D_{new}$

★ Attribute نوع مسابقت

$$D_{سه} = [0F, 2N] \rightarrow E = 0$$

$$D_{گرفتگی} = [2F, 1N] \rightarrow E = 0.9183$$

$$D_{درست است} = [0F, 0N] \rightarrow E = 0$$

$$\Rightarrow Gain(D_{new}, \text{نوع مسابقت}) = 0.971 - \frac{3}{5} \times 0.9183$$

$$\Rightarrow Gain(D_{new}, \text{نوع مسابقت}) = 0.42$$



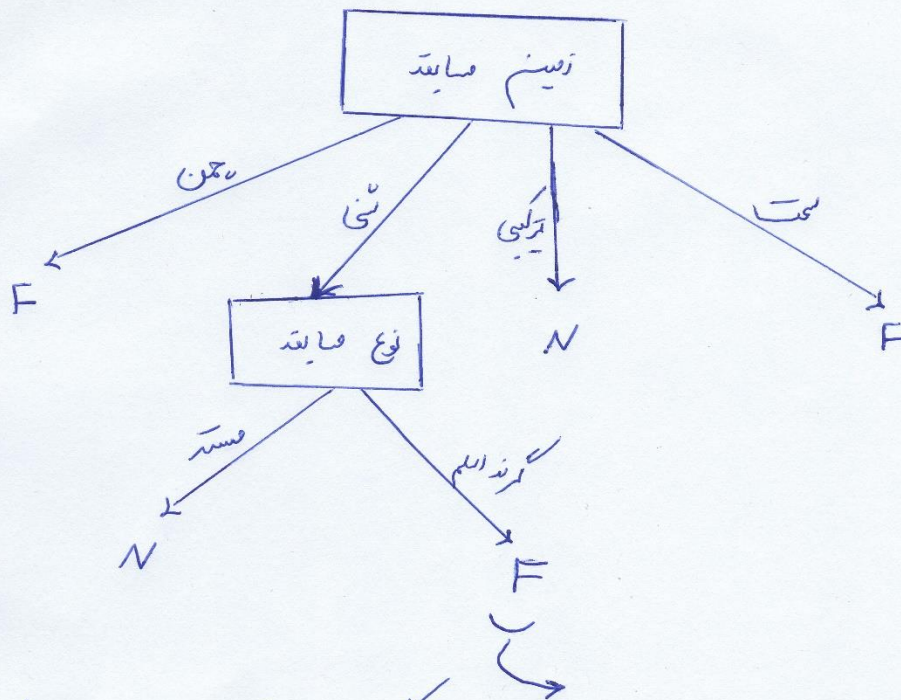
★ Attribute s حداکثر قدرت

$$D_1 = [2F, 3N] \rightarrow E = 0.971$$

$$D_0 = [0F, 0N] \rightarrow E = 0$$

$$\Rightarrow \text{Gain}(D_{\text{new}} \text{ و } \text{حداکثر قدرت}) = 0.971 - 0.971 = 0$$

$$\Rightarrow \begin{cases} \text{Gain}(D_{\text{new}}, \text{زمان}) = 0 \\ \text{Gain}(D_{\text{new}}, \text{نوع مایه}) = 0.42 \checkmark \\ \text{Gain}(D_{\text{new}}, \text{حداکثر قدرت}) = 0 \end{cases}$$



با توجه به اینکه احتمال پیروزی F در صورت بازی گرفت اسلم  $\frac{2}{3}$  و پیروزی N،  $\frac{1}{3}$  می باشد و با توجه به اینکه بودن attribute F در این زیر شاخه، گنجهایی که می شود لذا بیشترین احتمال پیروزی F را در نظر بگیرد



توجه نمایید که زیرشاخه‌ی دوستانه نیز چون در درخت تصمیم بدلیل عدم داده‌ی آموزش وجود ندارد لذا در صورت وجود آن در داده‌های تست، برچسب کلاس فوق را یعنی نوع مسابه را می‌گیرد و با توجه به اینکه احتمال نتیجه‌ی N برابر با 0.6 و نتیجه‌ی برابر با 0.4 است لذا برچسب کلاس نوع مسابقه یعنی N را به خود اختصاص می‌دهد.

(ب)

داده‌های تست را طبق صورت مسأله به صورت زیر در نظر داریم.

زمان	نوع مسابقه	زمین مسابقه	حداکثر قدرت	نتیجه
1 صبح	مستر	چمن	۱	F
2 بعدازظهر	گرنند اسلم	شنی	۱	N
3 بعدازظهر	مستر	ترکیبی	۰	F
4 صبح	مستر	شنی	۱	N
5 شب	دوستانه	سخت	۰	F
6 شب	گرنند اسلم	ترکیبی	۱	F

داده	پیش‌بینی شده بر اساس درخت تصمیم	نتیجه (برچسب واقعی)
داده‌ی تست 1	F	F
داده‌ی تست 2	F	N
داده‌ی تست 3	N	F
داده‌ی تست 4	N	N
داده‌ی تست 5	F	F
داده‌ی تست 6	N	F

$$Accuracy = \frac{\sum TP + TN}{\sum TP + FP + FN + TN} = \frac{3}{6} = 0.5 = 50\%$$

لذا accuracy با توجه به این مجموعه داده‌های test و train برابر با 50% می‌باشد.

## پاسخ سوال 4

الگوریتم boosting در واقع یک الگوریتم یادگیری برای مجموعه داده‌های supervised است که از نتایج کلاسیفرهای مختلف استفاده کرده و آنها را با یکدیگر ترکیب کرده و به طور کلی می‌توان گفت که با ترکیب سیستم‌های یادگیری ضعیف، یک سیستم یادگیری قوی را ایجاد می‌کند.

مبنای این الگوریتم بدین صورت است که یادگیرنده‌های ضعیف را به صورت iterative آموزش داده تا بدین صورت به یک کلاسیفایر قوی که ترکیب وزن‌داری از یادگیرنده‌های ضعیف و البته نمونه‌ها (داده‌های آموزش) هستند را تشکیل دهد. در واقع این الگوریتم در هر iteration با توجه به دقتی که یادگیرنده‌های ضعیف دارند به آنها وزن‌دهی کرده و در iteration های بعدی این وزن‌ها را آپدیت می‌کند و همچنین برای داده‌های آموزش نیز آنهایی که صحیح تشخیص داده شوند وزن کمتری در مجموعه داده‌ها گرفته و الگوریتم سعی می‌کند بر روی داده‌هایی که با آن چالش دارد تمرکز کرده و در نهایت با اینکار واضح است که دقت الگوریتم را در کلاس بندی بالا می‌برد.

Gradient boosting نیز مانند سایر الگوریتم‌های بوستینگ از ترکیب یادگیرنده‌های (یا مدل‌های) ضعیف است. در این الگوریتم از یک تابع ضرر برای بهینه‌سازی استفاده می‌کنیم مثلاً می‌توانیم از LSE (Least square error) استفاده کنیم. روند این الگوریتم بدین صورت است که سعی می‌کند با اضافه کردن یادگیرنده‌ها (مدل‌ها) جدید، این تابع ضرر را کمینه نماید و یک بهینه‌سازی انجام دهد. لذا این روند به صورت متناوب انجام می‌شود و نتیجتاً مدل‌ها (یادگیرنده‌های جدید) ساخته می‌شوند (یادگیرنده‌های ضعیف در اینجا درختانند تصمیم هستند). حال این مدل‌های جدید باید از خروجی مدل با تخمین مدل قبلی ارائه شود که این تخمین مدل در واقع با استفاده از گرادیان خطایی که ذکر کردیم بدست می‌آید. این خطا طبیعتاً بر روی مدل فعلی و مدل تخمینی که در مرحله قبل داشتیم محاسبه می‌شود و لذا ما به دنبال مدلی برای ارتقای تخمین به یادگیرنده‌ی قوی‌تر هستیم که بتواند منفی گرادیان خطا را برای ما مدل کند و با تجمیع کردن آن با مدل قبلی، یک مدل بهتر ایجاد کند. و مدام این روند را بر روی مدل واقعی و مدل تخمینی که در مرحله قبل بدست آمده اجرا کرده و خطای تخمین را بدست آورده و مدل بعدی با یک یادگیرنده‌ی جدید به گونه‌ای تخمین می‌زنیم که خطا را مینیمم کند.

## پاسخ سوال 5

به طور کلی هرس کردن یا pruning ابزار بسیار قوی برای کاهش complexity کلاسیفایر نهایی است و در واقع باعث بهبود صحت تخمین بوسیله‌ی کاهش اندازه‌ی درخت تصمیم (شاخه‌هایی که اطلاعات کمی دارند) و جلوگیری از overfitting داده‌ها می‌شود.

همانطور که در صورت سوال نیز در نوع از روش‌های موجود برای اینکار را ذکر کرده: 1. Pre pruning و 2. Post pruning

همانطو که از اسم این دو روش مشخص است می‌توانیم بگوییم که در pre pruning همچنانکه درخت ساخته می‌شود (در واقع قبل از ساخت نهایی decision tree) شاخه‌هایی تولیدی در صورتی که حاوی اطلاعات حائز اهمیتی و نباشند (یا به عبارتی non-significant) باشد را هرس کرده و از تولید این شاخه‌های جدید جلوگیری می‌کند و بدین صورت با توجه به شرایطی که به آن داده می‌شود اندازه‌ی درخت تصمیم را در حین ساخت درخت کاهش می‌دهد. ولی در post pruning پس از آنکه درخت تصمیم ساخته شد، شاخه‌هایی را که اطلاعات حائز اهمیتی ندارند را از درخت حذف یا به عبارتی هرس می‌کند. در واقع در این نوع هرس سازی در هر گام cross validation انجام شده و می‌بیند که آیا با بودن این شاخه accuracy افزایش می‌یابد یا خیر و در صورت اینکه صحت افزایش نیابد دیگه این شاخه در درخت قرار نخواهد گرفت و نقطه‌ی انتهایی زیر شاخه‌ی درخت یعنی برگ خواهد شد. مزیت و معایب هر یک به صورت زیر است.

### Pre pruning

✓ مزیت‌های این الگوریتم آن است که این الگوریتم یک نگاه رو به جلو دارد و در صورتیکه یک شاخه هرس شود دیگر تقسیم آن نود به زیرشاخه‌های کوچکتری متوالی که در زمان بدون هرس نیاز بود ایجاد شوند را نخواهد داشت و به گونه‌ای می‌توان از نظر زمانی هم (Time Saving) این الگوریتم را بدلیلی که ذکر کردیم هم مناسب دانست. همچنین از مزیت‌های دیگری که در متن هم ذکر نمودیم تولید مدلی ساده و البته دقیق و جلوگیری از overfitting داده هاست.

✓ عیب این الگوریتم این است که یافتن معیار هرس یا در واقع threshold مناسب برای اینکه عملیات pruning بهینه کار کند، دشوار است. همچنین این الگوریتم وابستگی زیادی به انتخاب داده‌های آموزش دارد و این داده‌ها اگر مناسب انتخاب نشوند مدل به خوبی هرس نخواهد شد و مدل خوبی را نتیجه نخواهد داد. و در کل rule های انتخابی هم حائز اهمیت



است و یافتن قوانین مناسب برای هرس کردن نیز از جمله معایب این روش خواهد بود. لذا در بعضی مواقع ممکن است underfitting رخ دهد که مطلوب نیست

### Post pruning

✓ از جمله مزایای این روش نیز مانند قبلی Time saving است هر چند ممکن است کمی زمان بیشتری را نسبت به الگوریتم قبلی مصرف کند و از مزیت دیگر این الگوریتم هرس کردن، سادگی روش است.

✓ معایب نیز این است که این الگوریتم نیز تعداد مناسب قواعد را بررسی نکرده و همچنین ممکن است در تولید قوانین، قوانین بهینه‌ای را اعمال نکند (دشواری در انتخاب قوانین) و به گونه‌ای قوانین آسیب پذیر به ساخت مدل بهتر منجر نشود. اعمال قوانین نامناسب و پارامترهای اولیه نامناسب در این الگوریتم نیز منجر به overfitting خواهد شد.

رفرنس : این بخش را علاوه بر اسلایدها در اینترنت نیز سرچ نمودیم و از مقاله‌ی زیر نیز استفاده کردم.

Rule Pruning Techniques in the Ant-Miner Classification Algorithm and Its Variants: A Review,  
Hayder Naser Khraibet AL-Behadili , Ku Ruhana Ku-Mahamud , Rafid Sagban

## پاسخ سوال 6

نرمالیزه کردن داده‌ها در جاهای مختلف کاربرد های مختلفی دارد. مثلاً یکی از کاربردهای برای زمانی است که داده‌های تکراری را در دیتابیس در بیش از یک جدول ذخیره می‌کنیم و باید آن‌ها را حذف کنیم که به عبارتی بیانگر redundancy در داده است که نرمالیزه کردن به راحتی به تشخیص و حذف اینگونه داده‌ها کمک می‌کند. همچنین علاوه بر کاهش افزونگی یا redundancy به کاهش پیچیدگی نیز کمک خواهد کرد. از دیگر اهداف نرمالیزه کردن تقسیم یک جدول بزرگ در دیتابیس به جداول کوچکی است که بین آنها ارتباطاتی به شکل منطقی و موثر وجود دارد. همچنین با نرمالیزه کردن می‌توان داده‌هایی را که در جداول اینگونه ارتباطات منطقی را با سایر داده‌ها ندارند حذف نمود و تنها داده‌هایی که با یکدیگر یک ارتباط منطقی و به نوعی dependency دارند را در جداول ذخیره نمود. این موارد اهداف کلی‌ای بود که برای نرمالیزه کردن می‌توان نام برد ولی توجه داشته باشید که نرمالیزه کردن می‌تواند در کاربردهای دیگر، نقش‌های موثر دیگری را نیز ایفا کند به عنوان مثال در شبکه‌های MLP ، نرمالیزه کردن داده‌ها باعث جلوگیری از گیر افتادن پاسخ (تنظیم وزن‌هایی که منجر به کاهش ارور می‌شوند) در بهینه‌های محلی می‌شود.

## پاسخ سوال 7

خیر ID3 تضمین نمی‌کند که به جوابی برسد که globally optimum باشد. این الگوریتم رویکرد حریصانه greedy دارد و لذا می‌دانیم که الگوریتم‌های حریصانه بر پایه‌ی جستجوی بهینه‌ی محلی با هدف رسیدن به globally optimum هستند ولی در بیشتر مواقع الگوریتم‌های حریصانه نمی‌توانند به این globally optimum برسند لذا به طور کلی می‌توان گفت که یک استراتژی حریصانه به طور معمول حل بهینه را به عنوان جواب برنمی‌گرداند با این حال اینگونه الگوریتم‌ها می‌توانند به یک حل بهینه‌ی محلی که البته تقریب خوبی از globally optimum هم باشد، در یک زمان منطقی برسند. لذا چون ID3 با این رویکرد به حل مسأله می‌پردازد، به همین جهت می‌گوییم که لزوماً این الگوریتم به globally optimum نمی‌رسد.

البته ناگفته نماند که بهینگی الگوریتم را می‌توان در طول فرایند جستجو برای رسیدن به درختی بهینه با استفاده از backtracking بهبود بخشید؛ هر چند که در قبال آن طبیعتاً پیچیدگی زمانی افزایش خواهد یافت.

## پاسخ سوال 8

→ ابتدا می‌دانیم accuracy, precision, recall به صورت زیر است:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \quad precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}$$

$$\Rightarrow accuracy = \frac{TP + TN}{(TP + TN) + (FP + FN)} = \frac{1}{1 + \frac{FP + FN}{TP + TN}}$$

$$= \frac{1}{1 + \left( \frac{FP + FN}{TP} \times \frac{TP}{TP + TN} \right)} = \frac{1}{1 + \left( \frac{FP + FN}{TP} \right) \times \left( 1 - \frac{TN}{TP + TN} \right)}$$

از طرف داریم :

$$\left\{ \begin{array}{l} \frac{1 - \text{Precision}}{\text{precision}} = \frac{\frac{FP}{TP+FP}}{\frac{TP}{TP+FP}} = \frac{FP}{TP} \\ \frac{1 - \text{recall}}{\text{recall}} = \frac{\frac{FN}{TP+FN}}{\frac{TP}{TP+FN}} = \frac{FN}{TP} \end{array} \right\} \Rightarrow \frac{1 - \text{Precision}}{\text{precision}} + \frac{1 - \text{recall}}{\text{recall}} = \frac{FP+FN}{TP}$$

$$\Rightarrow \text{Accuracy} = \frac{1}{1 + \left( \frac{1 - \text{precision}}{\text{precision}} + \frac{1 - \text{recall}}{\text{recall}} \right) \left( 1 - \frac{TN}{TP+TN} \right)}$$

و

$$\text{Accuracy} = \frac{1}{1 + \left( \frac{1 - \text{precision}}{\text{precision}} + \frac{1 - \text{recall}}{\text{recall}} \right) \left( \frac{TP}{TP+TN} \right)}$$

با تشکر - بدیعی