سوال ۱)

الف)

این احتمال وابسته به داده های آموزش می‌باشد. با توجه به داده ها احتمال spam بودن برابر $\frac{2}{3}$ می‌باشد. پس این مقدار دقیق نیست. اما درمورد اینکه منطقی است یا خیر، باید توجه کرد که داده های آموزش کم هستند و اگر کاملا مبنی را بر این مجموعه بگذاریم، احتمال خطای زیاد وجود دارد. همچنین درمورد مسئله‌ی تشخیص spam هزینه‌ی FP خیلی زیاد هست، در نتیجه اگر احتمال spam بودن کمتر درنظر گرفته شود و تعدادی از ایمیل ها را spam تشخیص ندهد بهتر از این است که به اشتباه spam تشخیص دهد. از لحاظ منطقی هم معمولا احتمال اسپم بودن کمتر است.

ب)

$$\mathrm{P(spam)} = 0.1, \mathrm{P(regular)} = 0.9$$

$$\mathrm{s} = (\mathrm{money} = 1, \mathrm{study} = 1, free = 0)$$

$$\mathrm{P(spam|s)} = \frac{\mathrm{P(s|spam)P(spam)}}{\mathrm{P(s)}}$$

این جمله با توجه به ویژگی های ما فقط شامل money و study و free ندارد. این احتمال ها را مستقل درنظر گرفته و داریم:

$$\mathrm{P(s|spam)P(spam)} = \frac{1}{8} \times \frac{4}{8} \times \frac{1}{8} \times \frac{1}{10}$$

برای spam نبودن هم محاسبه می‌کنیم:

$$\mathrm{P(regular|s)} = \frac{\mathrm{P(s|regular)P(regular)}}{\mathrm{P(s)}}$$

$$\mathrm{P(s|regular)P(regular)} = \frac{3}{4} \times \frac{1}{4} \times \frac{3}{4} \times \frac{9}{10}$$

P(s) که برابر است و می‌بینیم که احتمال اینکه regular باشد خیلی بیشتر است و بنابراین در این دسته قرار می‌گیرد.

سوال ۲)

در Decision tree ما attribute هایی را انتخاب می‌کنیم که بیشترین gain را داشته باشند.

حال برای تشخیص اینکه کدام کات برای جداسازی یک متغیر پیوسته بهتر است، با استفاده از الگوریتم C4.5 و روش MDLPC، میایم مقدار info را به ازای هر نقطه شکست پیدا می‌کنیم و آنکه info کمتری دارد را انتخاب می‌کنیم. درواقع آن که gain بیشتری را حاصل می‌کند.

محاسبه gain برای نقطه شکست اول:

$$\text{Entropy(D)} = -\frac{9}{16}\log\left(\frac{9}{16}\right) - \frac{7}{16}\log\left(\frac{7}{16}\right) = 0.9886$$

$$\text{Entropy}(\text{D}_{1=\text{left}}) = 0$$

$$\text{Entropy}\left(\text{D}_{1=\text{right}}\right) = -\frac{7}{14}\log\left(\frac{7}{14}\right) - \frac{7}{14}\log\left(\frac{7}{14}\right) = 1$$

$$\text{Gain(D, 1)} = \text{Entropy(D)} - \frac{2}{16}\text{Entropy}(\text{D}_{1=\text{left}}) - \frac{14}{16}\text{Entropy}\left(\text{D}_{1=\text{right}}\right)$$
$$= 0.9886 - \frac{2}{16} * 0 - \frac{14}{16} * 1 = 0.1136$$

محاسبه gain برای نقطه شکست دوم:

$$\text{Entropy(D)} = -\frac{9}{16}\log\left(\frac{9}{16}\right) - \frac{7}{16}\log\left(\frac{7}{16}\right) = 0.9886$$

$$\text{Entropy}(\text{D}_{2=\text{left}}) = -\frac{6}{9}\log\left(\frac{6}{9}\right) - \frac{3}{9}\log\left(\frac{3}{9}\right) = 0.9182$$

$$\text{Entropy}\left(\text{D}_{2=\text{right}}\right) = -\frac{6}{7}\log\left(\frac{6}{7}\right) - \frac{1}{7}\log\left(\frac{1}{7}\right) = 0.5916$$

$$\text{Gain(D, 2)} = \text{Entropy(D)} - \frac{9}{16}\text{Entropy}(\text{D}_{2=\text{left}}) - \frac{7}{16}\text{Entropy}\left(\text{D}_{2=\text{right}}\right)$$
$$= 0.9886 - \frac{9}{16} * 0.9182 - \frac{7}{16} * 0.5916 = 0.2132$$

همینطور که مشاهده می‌شود، نقطه شکست دوم مناسب‌تر می‌باشد.

سوال ۳)

الف)

$$\text{Entropy}(D) = -\frac{10}{15}\log\left(\frac{10}{15}\right) - \frac{5}{15}\log\left(\frac{5}{15}\right) = 0.918$$

محاسبه gain برای زمان:

$$\text{Entropy}(D_{\text{time=morning}}) = 0$$

$$\text{Entropy}(D_{\text{time=afternoon}}) = -\frac{6}{10}\log\left(\frac{6}{10}\right) - \frac{4}{10}\log\left(\frac{4}{10}\right) = 0.97$$

$$\text{Entropy}(D_{\text{time=night}}) = -\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right) = 0.918$$

$$\begin{aligned}\text{Gain}(D, \text{time}) &= \text{Entropy}(D) - \frac{2}{15}\text{Entropy}(D_{\text{time=morning}}) \\ &- \frac{10}{15}\text{Entropy}(D_{\text{time=afternoon}}) - \frac{3}{15}\text{Entropy}(D_{\text{time=night}}) \\ &= 0.918 - \frac{2}{15} * 0 - \frac{10}{15} * 0.97 - \frac{3}{15} * 0.918 = 0.0877\end{aligned}$$

محاسبه gain برای نوع مسابقه:

$$\text{Entropy}(D_{\text{type=master}}) = -\frac{3}{6}\log\left(\frac{3}{6}\right) - \frac{3}{6}\log\left(\frac{3}{6}\right) = 1$$

$$\text{Entropy}(D_{\text{type=grand}}) = -\frac{6}{7}\log\left(\frac{6}{7}\right) - \frac{1}{7}\log\left(\frac{1}{7}\right) = 0.591$$

$$\text{Entropy}(D_{\text{type=friendly}}) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$$

$$\begin{aligned}\text{Gain}(D, \text{type}) &= \text{Entropy}(D) - \frac{6}{15}\text{Entropy}(D_{\text{type=master}}) \\ &- \frac{7}{15}\text{Entropy}(D_{\text{type=grand}}) - \frac{2}{15}\text{Entropy}(D_{\text{type=friendly}}) \\ &= 0.918 - \frac{6}{15} * 1 - \frac{7}{15} * 0.591 - \frac{2}{15} * 1 = 0.1\end{aligned}$$

محاسبه gain برای زمین مسابقه:

$$\text{Entropy}(\text{D}_{\text{ground=grass}}) = 0$$

$$\text{Entropy}(\text{D}_{\text{ground=sand}}) = -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right) = 0.97$$

$$\text{Entropy}(\text{D}_{\text{ground=hard}}) = 0$$

$$\text{Entropy}(\text{D}_{\text{ground=mix}}) = 0$$

$$\begin{aligned}\text{Gain}(\text{D}, \text{ground}) &= \text{Entropy}(\text{D}) - \frac{4}{15}\text{Entropy}(\text{D}_{\text{ground=grass}}) \\ &- \frac{5}{15}\text{Entropy}(\text{D}_{\text{ground=sand}}) - \frac{4}{15}\text{Entropy}(\text{D}_{\text{ground=hard}}) \\ &- \frac{2}{15}\text{Entropy}(\text{D}_{\text{ground=mix}}) \\ &= 0.918 - \frac{4}{15} * 0 - \frac{5}{15} * 0.97 - \frac{4}{15} * 0 - \frac{2}{15} * 0 = 0.594\end{aligned}$$

محاسبه gain برای حداکثر قدرت:

$$\text{Entropy}(\text{D}_{\text{power=1}}) = -\frac{9}{13}\log\left(\frac{9}{13}\right) - \frac{4}{13}\log\left(\frac{4}{13}\right) = 0.89$$

$$\text{Entropy}(\text{D}_{\text{power=0}}) = 1$$

$$\begin{aligned}\text{Gain}(\text{D}, \text{power}) &= \text{Entropy}(\text{D}) - \frac{13}{15}\text{Entropy}(\text{D}_{\text{power=1}}) \\ &- \frac{2}{15}\text{Entropy}(\text{D}_{\text{power=0}}) = 0.918 - \frac{13}{15} * 0.89 - \frac{2}{15} * 1 \\ &= 0.0133\end{aligned}$$

طبق محاسبات انجام شده، ویژگی زمین مسابقه بیشترین مقدار gain را دارد و به عنوان اولین ویژگی انتخاب می‌شود. در مرحله بعد با توجه به نتایج بر اساس این ویژگی درخت را توسعه می‌دهیم.

مرحله دوم:

همانطور که مشاهده می‌شود، اگر زمین چمن و یا سخت باشد، در تمامی داده ها فدرر برنده شده است. اگر ترکیبی باشد در تمامی داده ها نادال برنده شده است. برای حالتی که زمین شنی باشد، مجدد gain ها را برای تصمیم‌گیری محاسبه می‌کنیم.

$$\text{Entropy}\left(\text{D}_{\text{ground=sand}}\right) = -\frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{3}{5}\log\left(\frac{3}{5}\right) = 0.97$$

محاسله gain برای زمان:

به ازای تمامی داده هایی که زمین شنی است، زمان نیز بعدازظهر است در نتیجه gain صفر می‌شود.

محاسله gain برای نوع مسابقه:

$$\text{Entropy}\left(\text{D}_{\text{ground=sand \& type=master}}\right) = 0$$

$$\text{Entropy}\left(\text{D}_{\text{ground=sand \& type=grand}}\right) = -\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right) = 0.918$$

$$\begin{aligned}\text{Gain}(\text{D}, \text{type}) &= \text{Entropy}(\text{D}) - \frac{2}{5}\text{Entropy}\left(\text{D}_{\text{ground=sand \& type=master}}\right) \\ &- \frac{3}{5}\text{Entropy}\left(\text{D}_{\text{ground=sand \& type=grand}}\right) \\ &= 0.97 - \frac{2}{5} * 0 - \frac{2}{3} * 0.918 = 0.358\end{aligned}$$

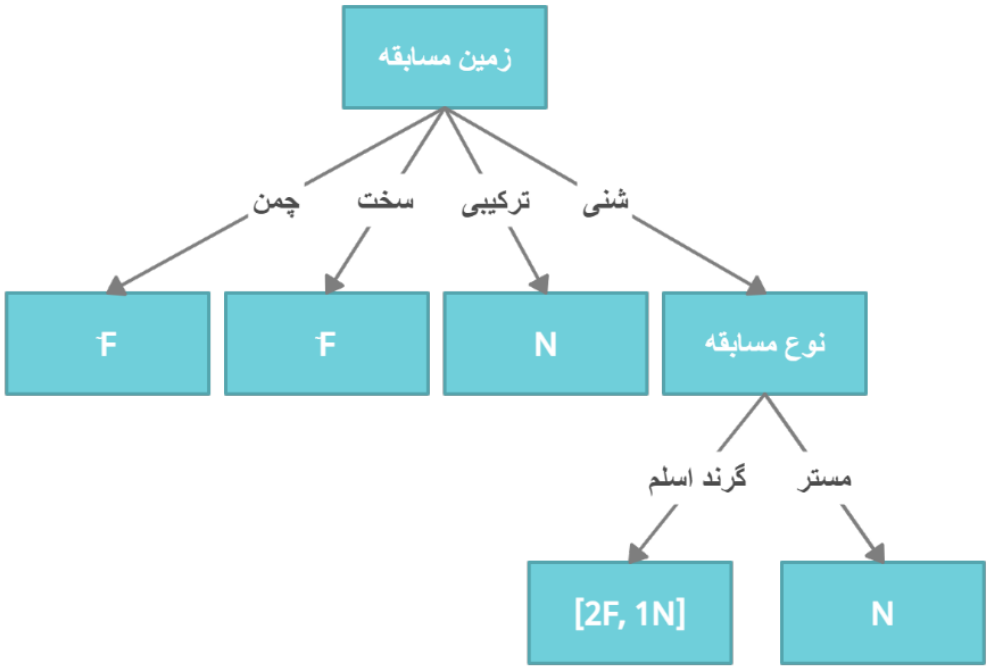
محاسله gain برای حداکثر قدرت:

به ازای تمامی داده هایی که زمین شنی است، حداکثر قدرت نیز ۱ است در نتیجه gain صفر می‌شود.

بنابراین ویژگی نوع مسابقه انتخاب می‌شود. حال داده ها را به ازای این ویژگی بررسی می‌کنیم.

با توجه به داده ها، اگر نوع مسابقه مستر باشد، نادال برنده می‌شود. و اگر نوع مسابقه گرند اسلم باشد، در دو تا از دیتا ها فدرر و یکی نادال برنده شده است. می‌بینیم در اینجا احتمال برد فدرر $\frac{2}{3}$ و نادال فدرر $\frac{1}{3}$ می‌باشد. اگر بخواهیم تصمیم قطعی بگیریم احتمال بیشتر یعنی فدرر را می‌توانیم درنظر بگیریم. همینطور برای زمانی که مثلا زمین شنی باشد و بازی دوستانه دیتایی دردسترس نداریم و می‌توان رندم نتیجه داد.

درخت تصمیم‌گیری بصورت زیر خواهد بود:



ب)

	نتیجه	حداکثر قدرت	زمین مسابقه	نوع مسابقه	زمان
F	F	۱	چمن	مستر	صبح
F	N	۱	شنی	گرند اسلم	بعدازظهر
N	F	۰	ترکیبی	مستر	بعدازظهر
N	N	۱	شنی	مستر	صبح
F	F	۰	سخت	دوستانه	شب
N	F	۱	ترکیبی	گرند اسلم	شب

می‌بینیم که سه مورد از شش مورد را درست تشخیص داد، در نتیجه خطا برابر 0.5 می‌باشد که بسیار زیاد است.

سوال ۴)

بطور کلی در روش boosting هدف ترکیب یادگیرنده های ضعیف و ایجاد یک یادگیرنده قوی است. مثلا در مسئله تشخیص spam بودن ایمیل قوانینی مانند یک URL خاص داشتن، شامل یک عکس خاص بودن، شامل کلید واژه های خاص و ... . این ها به تنهایی دقت خوبی ندارند و به آن ها یادگیرنده های ضعیف گفته می‌شود. در boosting با وزن های متفاوت نتایج هر یک از این قوانین را ترکیب می‌کنیم و یک یادگیرنده قوی می‌سازیم. مثلا اگر وزن های هر یک برابر باشد و نتایج دو تا spam باشد و یکی غیر spam، در نتیجه می‌گوییم spam است.

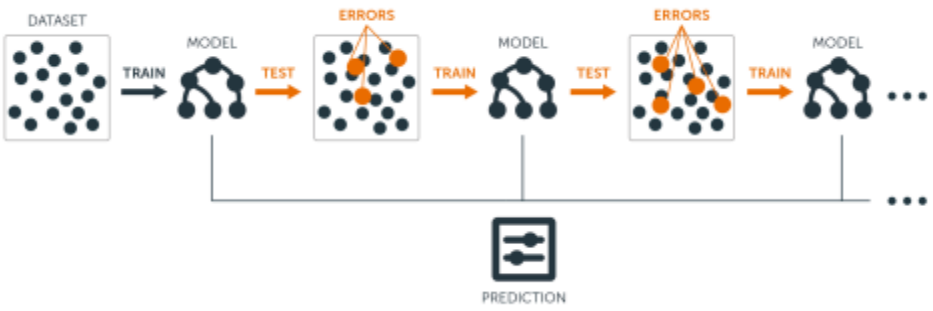
روند کلی این روش به این صورت است که بر اساس الگوریتم های ماشین لرنینگ و توزیع های مختلف در هر مرحله این قوانین و classifier ها ساخته شده و به مجموعه اضافه می‌شوند. هر یک از این از این یادگیرنده های ضعیف آموزش دیده و مدل های خود را می‌سازند. داده ها به این صورت داده می‌شود که اول به classifier سطح یک داده می‌شود و بعد بر اساس اینکه درست تشخیص دهد یا غلط وزن دهی می‌شوند(داده هایی که غلط تشخیص داده شوند، وزن بیشتری خواهند داشت) و به سطح بعدی داده می‌شوند.

Gradient Descent یکی از روش های boosting است که سه بخش اصلی دارد:

Loss function: اینکه چه تابعی استفاده شود بستگی به مدل دارد مثلا معمولا برای رگرسیون از مینیموم مربع خطا و برای کلاس بندی از هزینه لگاریتمی استفاده می‌شود.

Weak Learner: در این روش از درخت های تصمیم به عنوان یادگیرنده های ضعیف استفاده می‌شود.

Addictive Model: درخت های تصمیم را اضافه می‌کند و بعد از اضافه شدن هر کدام، از روش gradient descent استفاده می‌شود تا هزینه را مینیموم کند. در واقع این کار را با تخمین زدن اینکه کدام درخت با اضافه شدن هزینه را کاهش می‌دهد، انجام می‌دهد.



سوال ۵)

بطور کلی حرص کردن یک روش برای کاهش و فشرده سازی داده ها است و ایده این است که بخش هایی از درخت را که حیاتی نیستند و تاثیر زیادی ندارند، حذف کند. که به دو مدل pre-pruning و post-pruning تقصیم می‌شود.

Post-pruning: این روش بصورت backward است به اینصورت که پس از اینکه درخت ساخته شد، درخت ممکن از عمق زیادی داشته باشد و یکی دائم به شرط های کوچک‌تر شکسته شده باشد و باعث overfit شود، حال براساس استراتژی های مختلف مثلا minimum error که در آن درخت تا جایی که cross-validation مینیموم باشد، حرص می‌شود، و یا smallest tree.

Pre-pruning: این روش بصورت forward است و در زمان ساخت درخت، استفاده می‌شود و سعی می‌کند که پیش از اینکه درخت عمیق و باعث overfit شود، جلوی آن را بگیرد. به اینصورت که در هر مرحله cross-validation error را محاسبه می‌کند، و اگر کاهش قابل توجهی نسبت به مرحله قبل نداشته باشد، توسعه درخت را متوقف می‌کند.

مزایا و معایب:

Post-pruning سخت گیرانه تر عمل می‌کند و پیچیدگی بیشتری دارد و کند تر است، اما نتیجه آن قابل اعتماد تر است.

Pre-pruning: این مزیت را دارد که درخت عمیق نمی‌شود و یک هیوریستیک سریع به شمار می‌رود. اما انتخاب کردن یک threshold می‌تواند دشوار باشد همچنین ممکن است با حرص بیش از حد باعث underfitting شود.

سوال ۶)

هدف نرمال سازی داده ها در ماشین لرنینگ این است که یک scale مشترک برای ویژگی های عددی درنظر بگیرد. مثلا فرض کنیم دو ویژگی age که در بازه 0 تا 100 باشد و ویژگی درامد که در بازه 3 میلیون تا 30 میلیون باشد. این تفاوت شدید scale ها در مسائل مختلف می‌تواند مشکلاتی را ایجاد کند، مثلا در مسئله رگرسیون باعث می‌شود که ویژگی درآمد با توجه به اینکه مقدار عددی خیلی بیشتری دارد، خیلی تاثیر بیشتری داشته باشد و درنتیجه ممکن است به نتایج مطلوبی نرسیم.

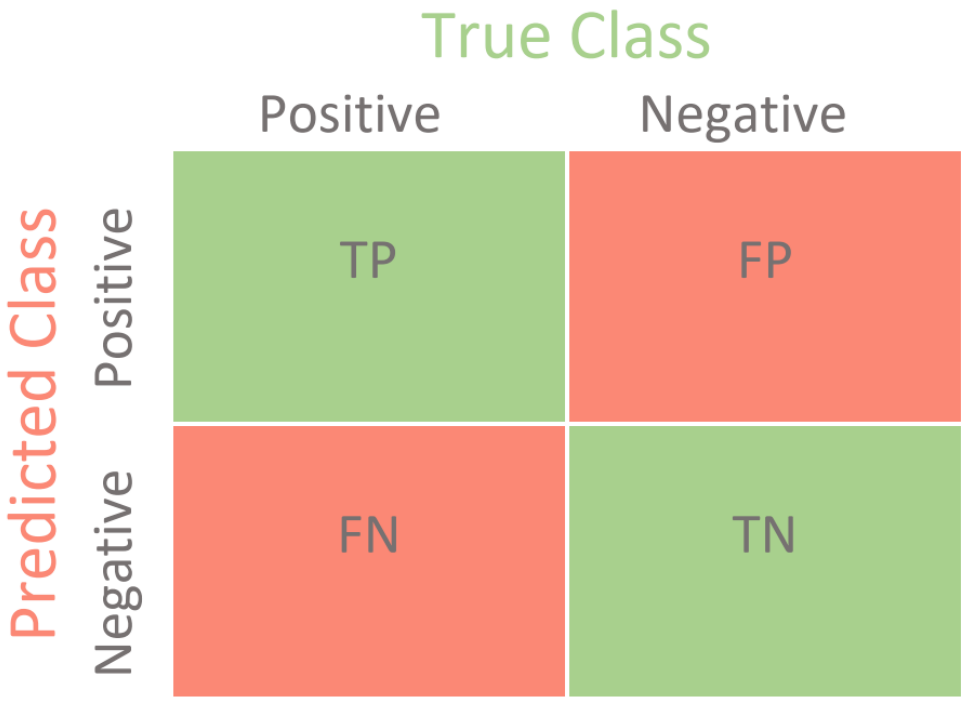
سوال ۷)

خیر، الگوریتم ID3 یک الگوریتم حریصانه است که در هر مرحله تصمیم می‌گیرد بهترین ویژگی را برای شکستن انتخاب کند. بنابراین می‌تواند در مینیموم های محلی گیر کند.

برای بهبود عملکرد این الگوریتم می‌توان از backtracking استفاده نمود.

سوال ۸)

ترم ها در ماتریکس بهم ریختگی بصورت زیر هستند:



true positives (TP)

زمانی که درست بوده اند و ما نیز YES پیش بینی کرده ایم.

true negatives (TN)

زمانی که درست نبوده اند و ما نیز NO پیش‌بینی کرده ایم.

false positives (FP)

زمانی که درست نبوده اند و ما YES پیش‌بینی کرده ایم.

false negatives (FN)

زمانی که درست بوده اند و ما NO پیش‌بینی کرده ایم.

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

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

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

$$\frac{1}{Precision} + \frac{1}{Recall} = \frac{TP + TP + FP + FN}{TP}$$

$$= 1 + \frac{TP + FP + FN + TN}{TP} - \frac{TN}{TP}$$

$$(\frac{1}{Precision} + \frac{1}{Recall} + \frac{TN}{TP} - 1)^{-1} = \frac{TP}{TP + FP + FN + TN}$$

$$Accuracy = (\frac{1}{Precision} + \frac{1}{Recall} + \frac{TN}{TP} - 1)^{-1} + \frac{TN}{TP + FP + FN + TN}$$