Revision for ML&DM module

<u>The in-class (online) test will include two parts. Part A will include 10 questions and you will be required to answer all of them. Part A contributes to 50% to the overall in-class test mark. Any topic covered in this module can be considered as a potential candidate for this part. Part B will include two simple case studies and you will be required to solve both of them using a specific algorithm/method. This part, obviously, contributes to the remaining 50% of the overall in-class test mark.</u>

The following lines illustrate again the problems shown in tutorial documents with the addition of some new problems.

What are z-score?

A z-score measures exactly how many standard deviations above or below the mean a data point is. Here's the formula for calculating a z-score:

$$z = \frac{\text{data point} - \text{mean}}{\text{standard deviation}}$$

Here's the same formula written with symbols:

$$z = \frac{x - \mu}{\sigma}$$

<u>Here are some important facts about z-scores:</u>

- A positive z-score says the data point is above average.
- A negative z-score says the data point is below average.
- A z-score close to 0 says the data point is close to average.

Suppose, we have the same 4 numbers: 8, 10, 15, 20, and we wish to find their z- score

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

$$\text{Standard deviation} = \sqrt{\frac{\sum(\text{every individual value of marks} - \text{mean of marks})^2}{n}}$$

Mean of marks = 8 + 10 + 15 + 20 /4 = 13.25

$$= \sqrt{\frac{(8 - 13.25)^2 + (10 - 13.25)^2 + (15 - 13.25)^2 + (20 - 13.25)^2}{4}}$$

$$= \sqrt{\frac{(-5.25)^2 + (-3.25)^2 + (1.75)^2 + (6.75)^2}{4}}$$

$$= \sqrt{\frac{27.56 + 10.56 + 3.06 + 45.56}{4}} = \sqrt{\frac{86.74}{4}} = \sqrt{21.6} = 4.6$$

$$ZScore = \frac{x-\mu}{\sigma} = \frac{8-13.25}{4.6} = -1.14$$

$$ZScore = \frac{x-\mu}{\sigma} = \frac{10-13.25}{4.6} = -0.7$$

$$ZScore = \frac{x-\mu}{\sigma} = \frac{15-13.25}{4.6} = 0.3$$

$$ZScore = \frac{x-\mu}{\sigma} = \frac{20-13.25}{4.6} = 1.4$$

Find all the eigenvalues for the given matrix

$$A = \begin{bmatrix} 4 & 0 & 1 \\ -1 & -6 & -2 \\ 5 & 0 & 0 \end{bmatrix}$$

Solution

$$\lambda I - A = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} - \begin{bmatrix} 4 & 0 & 1 \\ -1 & -6 & -2 \\ 5 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \lambda-4 & 0 & -1 \\ 1 & \lambda+6 & 2 \\ -5 & 0 & \lambda \end{bmatrix}$$

Now, let's take the determinant of this matrix and get the characteristic polynomial for A. We'll use the "trick" that we reviewed in the previous section to take the determinant. You could also use cofactors if you prefer that method. The result will be the same.

$$\det(\lambda I - A) = \left| \begin{matrix} \lambda-4 & 0 & -1 \\ 1 & \lambda+6 & 2 \\ -5 & 0 & \lambda \end{matrix} \right| \begin{matrix} \lambda-4 & 0 \\ 1 & \lambda+6 \\ -5 & 0 \end{matrix}$$

$$= \lambda(\lambda-4)(\lambda+6) - 5(\lambda+6)$$

$$= \lambda^3 + 2\lambda^2 - 29\lambda - 30$$

Next, set this equal to zero.

$$\lambda^3 + 2\lambda^2 - 29\lambda - 30 = 0$$

Now, most of us aren't that great at find the roots of a cubic polynomial. Luckily there is a way to at least get us started. It won't always work, but if it does it can greatly reduce the amount of work that we need to do.

Suppose we're trying to find the roots of an equation of the form,

$$\lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0 = 0$$

where the c_i are all integers. If there are integer solutions to this (and there may NOT be) then we know that they must be divisors of c_0. This won't give us any integer solutions, but it will allow us to write down a list of possible integer solutions. The list will be all possible divisors of c_0.

In this case the list of possible integer solutions is all possible divisors of -30.

$$\pm 1, \pm 2, \pm 3, \pm 5, \pm 6, \pm 10, \pm 15, \pm 30$$

Now, that may seem like a lot of solutions that we'll need to check. However, it isn't quite that bad. Start with the smaller possible solutions and plug them in until you find one (*i.e.* until the polynomial is zero for one of them) and then stop. In this case the smallest one in the list that works is -1. This means that

$$\lambda - (-1) = \lambda + 1$$

must be a factor in the characteristic polynomial. In other words, we can write the characteristic polynomial as,

$$\lambda^3 + 2\lambda^2 - 29\lambda - 30 = (\lambda + 1) q(\lambda)$$

where $q(\lambda)$ is a quadratic polynomial. We find $q(\lambda)$ by performing long division on the characteristic polynomial. Doing this in this case gives,

$$\lambda^3 + 2\lambda^2 - 29\lambda - 30 = (\lambda + 1)(\lambda^2 + \lambda - 30)$$

At this point all we need to do is find the solutions to the quadratic and nicely enough for us that factors in this case. So, putting all this together gives,

$$(\lambda + 1)(\lambda + 6)(\lambda - 5) = 0 \qquad \Rightarrow \qquad \lambda_1 = -1,\ \lambda_2 = -6,\ \lambda_3 = 5$$

So, this matrix has three simple eigenvalues.

Clustering

For three objects, A: (1, 0, 1, 1), B: (2, 1, 0, 2) and C: (2, 2, 2, 1), store them in a data matrix and use Manhattan and Euclidean distances to generate distance matrices, respectively

The data matrix should be $\begin{bmatrix} 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 \\ 2 & 2 & 2 & 1 \end{bmatrix}$.

Using the Manhattan distance, we have

$$d(A,B) = |1-2|+|0-1|+|1-0|+|1-2| = 1+1+1+1 = 4$$
$$d(A,C) = |1-2|+|0-2|+|1-2|+|1-1| = 1+2+1+0 = 4$$
$$d(B,C) = |2-2|+|1-2|+|0-2|+|2-1| = 0+1+2+1 = 4$$

The Manhattan distance matrix is $\begin{bmatrix} 0 & 4 & 4 \\ 4 & 0 & 4 \\ 4 & 4 & 0 \end{bmatrix}$.

Using the Euclidean distance, we have

$$d(A,B) = \sqrt{(1-2)^2 + (0-1)^2 + (1-0)^2 + (1-2)^2} = \sqrt{1+1+1+1} = 2$$
$$d(A,C) = \sqrt{(1-2)^2 + (0-2)^2 + (1-2)^2 + (1-1)^2} = \sqrt{1+4+1+0} = \sqrt{6}$$
$$d(B,C) = \sqrt{(2-2)^2 + (1-2)^2 + (0-2)^2 + (2-1)^2} = \sqrt{0+1+4+1} = \sqrt{6}$$

The Euclidean distance matrix is $\begin{bmatrix} 0 & 2 & \sqrt{6} \\ 2 & 0 & \sqrt{6} \\ \sqrt{6} & \sqrt{6} & 0 \end{bmatrix}$.

Given a one-dimensional data set {2, 4, 5, 9, 10}, it has been divided into two clusters {1, 2, 3} and {4, 5}, use single, complete and average links with Euclidean distance to calculating the distances between them, respectively.

We first calculate the pairwise data distance with Euclidean distance to find the distance matrix of this data set as follows:

$$\begin{bmatrix} 0 & 2 & 3 & 7 & 8 \\ 2 & 0 & 1 & 5 & 6 \\ 3 & 1 & 0 & 4 & 5 \\ 7 & 5 & 4 & 0 & 1 \\ 8 & 6 & 5 & 1 & 0 \end{bmatrix}$$

For the single link, we need to find the shortest distance; i.e.,

d({1, 2, 3}, {4, 5}) = min{ d(1,4), d(1, 5), d(2,4), d(2,5), d(3,4), d(3,5)}
= min{ 7, 8, 5, 6, 4, 5}
= 4

For the complete link, we need to find the longest distance; i.e.,

d({1, 2, 3}, {4, 5}) = max{ d(1,4), d(1, 5), d(2,4), d(2,5), d(3,4), d(3,5)}
= max{ 7, 8, 5, 6, 4, 5}
= 8

For the average link, we need to find the averaging distance; i.e.,

d({1, 2, 3}, {4, 5}) = [d(1,4) + d(1, 5) + d(2,4) + d(2,5) + d(3,4) + d(3,5)}]/6
= [7 + 8 + 5 + 6 +4 + 5]/6
= 35/6

K-means clustering

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). The distance matrix based on the Euclidean distance is given below:

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Suppose that the initial seeds (centres of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show:

a) The new clusters (i.e. the examples belonging to each cluster)

b) The centres of the new clusters

a)

d(a,b) denotes the Eucledian distance between a and b. It is obtained directly from the distance matrix or calculated as follows: d(a,b)=sqrt($(x_b-x_a)^2+(y_b-y_a)^2$))

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

epoch1 – start:

A1:

d(A1, seed1)=0 as A1 is seed1

d(A1, seed2)= $\sqrt{13}$ >0

d(A1, seed3)= $\sqrt{65}$ >0

➔A1 $\in$ cluster1

A2:

d(A2,seed1)= $\sqrt{25}$ = 5

d(A2, seed2)= $\sqrt{18}$ = 4.24

d(A2, seed3)= $\sqrt{10}$ = 3.16 ← smaller

➔ A2 $\in$ cluster3

A3:
d(A3, seed1)= $\sqrt{36}$ = 6
d(A3, seed2)= $\sqrt{25}$ = 5 ← smaller
d(A3, seed3)= $\sqrt{53}$ = 7.28
➔ A3 ∈ cluster2

A4:
d(A4, seed1)= $\sqrt{13}$
d(A4, seed2)=0 as A4 is seed2
d(A4, seed3)= $\sqrt{52}$ >0
➔ A4 ∈ cluster2

A5:
d(A5, seed1)= $\sqrt{50}$ = 7.07
d(A5, seed2)= $\sqrt{13}$ = 3.60 ← smaller
d(A5, seed3)= $\sqrt{45}$ = 6.70
➔ A5 ∈ cluster2

A6:
d(A6, seed1)= $\sqrt{52}$ = 7.21
d(A6, seed2)= $\sqrt{17}$ = 4.12 ← smaller
d(A6, seed3)= $\sqrt{29}$ = 5.38
➔ A6 ∈ cluster2

A7:
d(A7, seed1)= $\sqrt{65}$ >0
d(A7, seed2)= $\sqrt{52}$ >0
d(A7, seed3)=0 as A7 is seed3
➔ A7 ∈ cluster3

A8:
d(A8, seed1)= $\sqrt{5}$
d(A8, seed2)= $\sqrt{2}$ ← smaller
d(A8, seed3)= $\sqrt{58}$
➔ A8 ∈ cluster2

end of epoch1

new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

b) centers of the new clusters:
C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)

Assume the following dataset is given: (2,2), (4,4), (5,5), (6,6), (9,9), (0,4), (4,0). K-Means is run with k=3 to cluster the dataset. Moreover, Manhattan distance is used as the distance function to compute distances between centroids and objects in the dataset. In addition, K-Mean's initial clusters C1, C2, and C3 are as follows:
C1: {(2, 2), (4, 4), (6, 6)}
C2: {(0, 4), (4, 0)}
C3: {(5, 5), (9, 9)}
We need to run K-means is run for a single iteration; what are the new clusters and what are their new centroids?

Remember that this distance is: d((x1,x2),(x1',x2'))= |x1-x1'| + |x2-x2'|

Current centroids: Center c1: (4,4) Center c2: (2,2) Center c3: (7,7)

d(2,2)(4,4)=4;	d(2,2)(2,2)=0;	d(2,2)(7,7)=10;
d(4,4)(4,4)=0;	d(4,4)(2,2)=4;	d(4,4)(7,7)=6;
d(5,5)(4,4)=2;	d(5,5)(2,2)=6;	d(5,5)(7,7)=4;
d(6,6)(4,4)=4;	d(6,6)(2,2)=8;	d(6,6)(7,7)=2;
d(9,9)(4,4)=10;	d(9,9)(2,2)=14;	d(9,9)(7,7)=4;
d(0,4)(4,4)=4;	d(0,4)(2,2)=4;	d(0,4)(7,7)=10;
d(4,0)(4,4)=4;	d(4,0)(2,2)=4;	d(4,4)(7,7)=10;

So:
c1{(4,4),(5,5),(0,4),(4,0)}, or c1{(4,4),(5,5)} or…
c2{(2,2)} or c2{(2,2), (0.4),(4.0)}
c3{(6,6),(9,9)}
Calculate the new centroids now. What is the difference if we had to apply Euclidean distance?

Hierarchical clustering

Use single and complete link agglomerative clustering to group the data described by the following distance matrix. Show the related dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

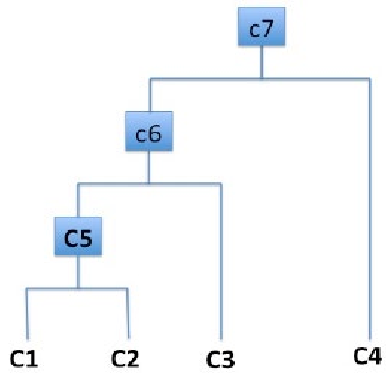
Single link: distance between two clusters is the shortest distance between a pair of elements from the two clusters.

We apply the algorithm presented in lecture 10 (ml_2012_lecture_10.pdf), page 4.

At the beginning, each point A,B,C, and D is a cluster → c1 = {A}, c2={B}, c3={C}, c4={D}

Iteration 1
The shortest distance is d(c1,c2)=1 → c1 and c2 are merged → the clusters are c3={C}, c4={D}, c5={A,B}
The distances from the new cluster to the others are d(c5,c3) = 2, d(c5,c4)=5
Iteration 2
The shortest distance is d(c5,c3)=2 → c5 and c3 are merged → the clusters are c6={A,B,C}, c4={D}
The distances from the new cluster to the others are: d(c6,c4)=3

Iteration 3
c6 and c4 are merged → the final cluster is c7={A,B,C,D}



Complete link: The distance between two clusters is the distance of two furthest data points in the two clusters
We apply the algorithm presented in lecture 10 (ml_2012_lecture_10.pdf) page 4.

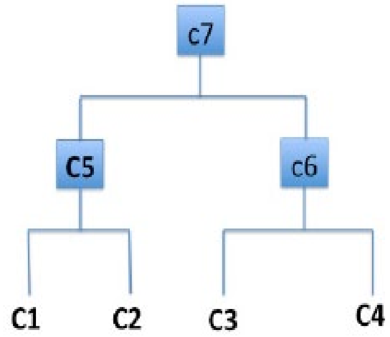
At the beginning, each point A,B,C, and D is a cluster → c1 = {A}, c2={B}, c3={C}, c4={D}

Iteration 1
The shortest distance is d(c1,c2)=1 → c1 and c2 are merged → the clusters are c3={C}, c4={D}, c5={A,B}
The distances from the new cluster to the others are: d(c5,c3) = 4, d(c5,c4)=6

Iteration 2
The shortest distance is d(c3,c4)=3 → c3 and c4 are merged → the clusters are c6={C,D}, c5={A,B}
The distances from the new cluster to the others are: d(c6,c5)=6

Iteration 3
c6 and c5 are merged → the final cluster is c7={A,B,C,D}

The dendrogram is



Given a one-dimensional data set {1, 5, 8, 10, 2}, use the agglomerative clustering algorithm with the complete link and the Euclidean distance to establish a hierarchical grouping relationship.

Solution
In order to use the agglomerative algorithm, we need to calculate the distance matrix.

$$\begin{bmatrix} 0 & 4 & 7 & 9 & 1 \\ 4 & 0 & 3 & 5 & 3 \\ 7 & 3 & 0 & 2 & 6 \\ 9 & 5 & 2 & 0 & 8 \\ 1 & 3 & 6 & 8 & 0 \end{bmatrix}$$

From the distance matrix, we can find that the distance between points (i.e. positions) 1 and 5 is smallest. Therefore, we can merge them together with their distance as the threshold. Then, we update the distance matrix by using the cluster {1, 5}. Using the complete link, we can re-calculate the distance between this cluster and other points.

$d(2, \{1,5\}) = \max\{ d(2,1), d(2,5) \} = \max\{4, 3\} = 4$
$d(3, \{1,5\}) = \max\{ d(3,1), d(3,5) \} = \max\{7, 6\} = 7$
$d(4, \{1,5\}) = \max\{ d(4,1), d(4,5) \} = \max\{9, 8\} = 9$

Let the 1_{st} column (row) denote the distances between this cluster {1, 5} and other points, we have the following distance matrix:

$$\begin{bmatrix} 0 & 4 & 7 & 9 \\ 4 & 0 & 3 & 5 \\ 7 & 3 & 0 & 2 \\ 9 & 5 & 2 & 0 \end{bmatrix}$$

From the above distance matrix, we can see the distance between points 3 and 4 is smallest. Hence, they merge together to form a cluster {3, 4}. Using the complete link, we have the distance between different points/clusters as follows:

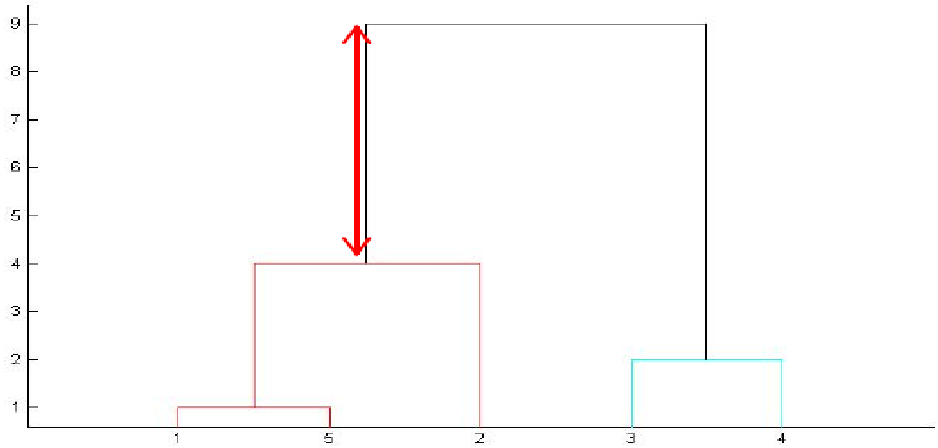
d({1,5}, {3, 4}) = max{ d({1,5}, 3), d({1,5}, 4) } = max{ 7, 9} = 9
d(2, {3,4}) = max{ d(2,3), d(2,4) } = max {3, 5} = 5
Thus, we can update the distance matrix, where row 2 corresponds to point 2, rows 1 and 3 correspond to clusters {1, 5} and {3, 4}, as follows:

$$\begin{bmatrix} 0 & 4 & 9 \\ 4 & 0 & 5 \\ 9 & 5 & 0 \end{bmatrix}$$

Following the same procedure, we merge point 2 with the cluster {1, 5} to form {1, 2, 5} and update the distance matrix as follows:

$$\begin{bmatrix} 0 & 9 \\ 9 & 0 \end{bmatrix}$$

After increasing the distance threshold to 9, all clusters would merge. Based on all above distance matrices, we draw the dendrogram tree as follows:



Naïve Bayes

Our very simple naïve Bayes problem. In this case, we have only one input variable/attribute. Normally we have more. We have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. The question is are players will play if weather is sunny?

Solution:
Step 1: Convert the data set into a frequency table
Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64
Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.
Let's see the opposite:
P(No | Sunny) = P(Sunny |No) * P(No) / P (Sunny)
Here we have P (Sunny |No) = 2/5 = 0.4, P(Sunny) = 5/14 = 0.36, P(No)= 5/14 = 0.36
Now, P (No | Sunny) = 0.4 * 0.36 / 0.36 = 0.40, which has lower probability.
Thus, the answer is yes, play.

Given the training data in the table below (Buy Computer data) predict the class (yes or no) of the following example using Naïve Bayes classification: age<=30, income=medium, student=yes, credit-rating=fair (this is our target)

Solution:

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

E= age<=30, income=medium, student=yes, credit-rating=fair
E_1 is age<=30, E2 is income=medium, student=yes, E4 is credit-rating=fair
We need to compute P(yes|E) and P(no|E) and compare them.

$$P(yes \mid E) = \frac{P(E_1 \mid yes)P(E_2 \mid yes)P(E_3 \mid yes)P(E_4 \mid yes)P(yes)}{P(E)}$$

P(yes)=9/14=0.643 P(no)=5/14=0.357

P(E1|yes)=2/9=0.222	P(E1|no)=3/5=0.6
P(E2|yes)=4/9=0.444	P(E2|no)=2/5=0.4
P(E3|yes)=6/9=0.667	P(E3|no)=1/5=0.2
P(E4|yes)=6/9=0.667	P(E4|no)=2/5=0.4

$$P(yes \mid E) = \frac{0.222\ 0.444\ 0.667\ 0.668\ 0.443}{P(E)} = \frac{0.028}{P(E)} \qquad P(no \mid E) = \frac{0.6\ 0.4\ 0.2\ 0.4\ 0.357}{P(E)} = \frac{0.007}{P(E)}$$

Hence, the Naïve Bayes classifier predicts buys_computer=yes for the new example.

Check: Here the denominator P(E) is not calculated, as it is the same in both cases, so does not have a practical effect in the result. How much is P(E) however? Consult the slides to find it out!

This weather dataset has 14 instances and five numbers of attributes. Here, first four attributes are predictors and the last attribute is the target attribute (if we have to play golf).

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.
See the following schematic, as an example:

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

$$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(No) = 5/14 = 0.36$$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

Now, back to our problem. The likelihood tables for all four predictors are:

Frequency Table / Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

In this example we have 4 inputs (predictors). The final posterior probabilities can be standardized between 0 and 1. **Our Target:** We need to classify the following new instance:
Outlook: Rainy, Temp: Cool, Humidity: High, Windy: True, Play:?

P(Yes|X) = P(Rainy|Yes) x P(Cool|Yes) x P(High|Yes) x P(True|Yes) x P(Yes)
P(Yes|X) = 2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.00529
or 0.00529/ (0.00529+ 0.02057) = 0.20 (this is an option)

P(No|X) = P(Rainy|No) x P(Cool|No) x P(High|No) x P(True|No) x P(No)
P(No|X) = 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.02057
or 0.02057/ (0.00529+ 0.02057) = 0.80

So the probability for no is highest as compare to yes, so it is more likely to watch a movie instead of playing golf!

Consider the following dataset:

N	Color	Type	Origin	Stolen?
1	red	sports	domestic	yes
2	red	sports	domestic	no
3	red	sports	domestic	yes
4	yellow	sports	domestic	no
5	yellow	sports	imported	yes
6	yellow	SUV	imported	no
7	yellow	SUV	imported	yes
8	yellow	SUV	domestic	no
9	red	SUV	imported	no
10	red	sports	domestic	yes

Classify the car with the specific characteristics (red, SUV, domestic) using the Naïve Bayes classifier. Is it stolen or not?

P(Stolen=yes)=1/2
P(Stolen=no)=1/2
P(red|Stolen=yes)=3/5
P(red|Stolen=no)=2/5
P(SUV|Stolen=yes)=1/5
P(SUV|Stolen=no)=3/5
P(domestic|Stolen=yes)=3/5
P(domestic|Stolen=no)=3/5

P(Stolen=yes|red, SUV, domestic)= P(red|Stolen=yes) * P(SUV|Stolen=yes)* P(domestic|Stolen = yes) * P(Stolen=yes)= 3/5*1/5*3/5*1/2 = 9/250.

P(Stolen=no|red, SUV, domestic)= P(red|Stolen=no)* P(SUV|Stolen=no) * P(domestic|Stolen=no)* P(Stolen=no) = 2/5*3/5*3/5*1/2=18/250
This car is safe (not stolen)

Association Rules

Consider a transactional database where 1, 2, 3, 4, 5, 6, 7 are items.

ID	Items
t 1	1, 2, 3, 5
t 2	1, 2, 3, 4, 5
t 3	1, 2, 3, 7
t 4	1, 3, 6
t 5	1, 2, 4, 5, 6

Suppose the minimum support is 60%. Find all frequent itemsets. Indicate each candidate set C_k, k = 1, 2,.., the candidates that are pruned by each pruning step, and the resulting frequent itemsets L_k.

Use Apriori algorithm to find all frequent itemsets. Itemsets shaded in grey are removed because they fail the minimum support constraint. Those shaded in light yellow are removed because there exists a subset of itemsets that is not frequent. Minimum support count = 5 × 60% = 3

C_1

Itemset	Support
{1}	5
{2}	4
{3}	4
{4}	2
{5}	3
{6}	2
{7}	1

→

L_1

Itemset	Support
{1}	5
{2}	4
{3}	4
{5}	3

C_2

Itemset	Support
{1, 2}	4
{1, 3}	4
{1, 5}	3
{2, 3}	3
{2, 5}	3
{3, 5}	2

→

L_2

Itemset	Support
{1, 2}	4
{1, 3}	4
{1, 5}	3
{2, 3}	3
{2, 5}	3

C_3

Itemset	Support
{1, 2, 3}	3
{1, 2, 5}	3
{1, 3, 5}	
{2, 3, 5}	

→

L_3

Itemset	Support
{1, 2, 3}	3
{1, 2, 5}	3

C_4

Itemset	Support
{1, 2, 3, 5}	

Let the minimum support be 60% and minimum confidence be 75%. Show all association rules that are constructed from the same transaction dataset.

All association rules generated from L_2 and L_3 are shown below together with support and confidence. All rows that are not shaded are association rules with confidence ≥ 75%.

Association Rule	Support	Confidence
{1} → {2}	4 (80%)	4/5 (80%)
{2} → {1}	4 (80%)	4/4 (100%)
{1} → {3}	4 (80%)	4/5 (80%)
{3} → {1}	4 (80%)	4/4 (100%)
{1} → {5}	3 (60%)	3/5 (60%)
{5} → {1}	3 (60%)	3/3 (100%)
{2} → {3}	3 (60%)	3/4 (75%)
{3} → {2}	3 (60%)	3/4 (75%)
{2} → {5}	3 (60%)	3/4 (75%)
{5} → {2}	3 (60%)	3/3 (100%)
{1} → {2, 3}	3 (60%)	3/5 (60%)
{2} → {1, 3}	3 (60%)	3/4 (75%)
{3} → {1, 2}	3 (60%)	3/4 (75%)
{1, 2} → {3}	3 (60%)	3/4 (75%)
{1, 3} → {2}	3 (60%)	3/4 (75%)
{2, 3} → {1}	3 (60%)	3/3 (100%)
{1} → {2, 5}	3 (60%)	3/5 (60%)
{2} → {1, 5}	3 (60%)	3/4 (75%)
{5} → {1, 2}	3 (60%)	3/3 (100%)
{1, 2} → {5}	3 (60%)	3/4 (75%)
{1, 5} → {2}	3 (60%)	3/3 (100%)
{2, 5} → {1}	3 (60%)	3/3 (100%)

Given the transaction database below

TID	Items
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

Apply the Apriori algorithm with minimum support count threshold 3 and find the set M of maximal frequent itemsets.

We need to find first the frequent itemsets in the form candidate & final lists. For example C_1 = {A: 4, …}, L_1={…}.

C1 = {A: 4, B: 6, C: 4, D: 4, E: 5}
L1 = {A, B, C, D, E}
C2 = {AB:4, AC:2, AD:3, AE:4, BC:4, BD:4, BE:5, CD:2, CE:3, DE:3}
L2 = {AB, AD, AE, BC, BD, BE, CE, DE}
C3 = {ABD: 3, ABE: 4, ADE: 3, BCD: 2, BCE: 3, BDE: 3}
L3 = {ABD, ABE, ADE, BCE, BDE}
C4 = {ABDE: 3}

L4 = {ABDE}
C5 = {}
L5 = {}

Maximal frequent itemset: The definition says that an itemset is maximal frequent if none of its immediate supersets is frequent.
From the itemsets of the previous list, we need to find those ones which have all of their immediate supersets infrequent.

M = {ABDE, BCE}
For example ABCE: 2, BCDE: 1, which means that all immediate supersets of BCE are infrequent, for this reason BCE is maximal. For ABDE there is no C5/L5, so also it is maximal.

Given the transaction database below

TID	Items
1	A,C,D,F
2	B,C,E
3	A,B,C,E
4	B,D,E
5	A,B,C,E
6	A,B,C,D

Apply the Apriori algorithm with minimum support count threshold 4 and find (a) the set of frequent itemsets, (b) the set of closed frequent itemsets and (c) the set of maximal frequent itemsets.

We check first the 1-size itemsets
A 4
B 5
C 5
D 3 reject due to less than 4
E 4
F 1

Only 4 1-temsets will be considered as candidates for the creation of 2-size itemsets
AB 3
AC 4
AE 2
BC 4
BE 4
CE 3

3 2-itemsets are suitable; they will create the 3-size itemsets in the next stage

ABC 3 and ab
ACE reject due to ae
ABE 2 and ae
BCE 3 and ce

Thus, none 3-itemsets are frequent.
Remember the definitions:

- **Closed Frequent Itemset:** An itemset is closed if none of its immediate supersets has the same support as that of the itemset.
- **Maximal frequent itemset:** The definition says that an itemset is maximal frequent if none of its immediate supersets is frequent

Let's try to find the closed frequent itemsets

B, C, BE, AC, BC

A and E have been rejected as their immediate supersets have the save support. The maximal ones are obviously
BE, AC, BC

Consider the market basket transactions shown in the following table. Assume that min_support=40%. Show step by step the generated candidate itemsets (C_k) and the qualified frequent itemsets (L_k) until the largest frequent itemset(s) are generated.

TID	Items
1	{Bread, Butter, Milk}
2	{Bread, Butter}
3	{Beer, Cookies, Diapers}
4	{Milk, Diapers, Bread, Butter}
5	{Beer, Diapers}

Do it as a practice. Just for your information, there are 5 L_1, 4 L_2 and 1 L_3 frequent itemsets.

Decision Trees

We have been given the following medical dataset and wish to create a decision tree (using the ID3 algorithm). The dataset includes four attributes. Calculate which attribute needs to be selected for the first (root) node. Justify your response.

Training	fever	vomiting	diarrhea	shivering	Classification
d_1	no	no	no	no	healthy (H)
d_2	average	no	no	no	influenza (I)
d_3	high	no	no	yes	influenza (I)
d_4	high	yes	yes	no	salmonella poisoning (S)
d_5	average	no	yes	no	salmonella poisoning (S)
d_6	no	yes	yes	no	bowel inflammation (B)
d_7	average	yes	yes	no	bowel inflammation (B)

For the calculation of $\log_2$, remember, from maths, that: $\log_a b = \frac{\log_{10} b}{\log_{10} a}$

$\log_2$ approx.	
log2(1/8)	= -3.0p
log2(1/4)	= -2.00
log2(1/3)	= -1.58
log2(3/8)	= -1.42
log2(3/7)	= -1.22
log2(1/2)	= -1.00
log2(4/7)	= -0.81
log2(5/8)	= -0.68
log2(2/3)	= -0.58
log2(3/4)	= -0.42
log2(7/8)	= -0.19

Solution

The parent entropy (i.e. total) is calculated as:

$$Entropy(S) = -\frac{1}{7}\log_2\left(\frac{1}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right) = 1.948$$

We need to check the entropy for each attribute individually and then find which attribute maximizes the information gain.

Fever attribute:

Values	H	I	S	B	Entropy(S_i)
S_1 (no)	x			x	[1/2,0,0,1/2]
S_2(average)		x	x	x	[0,1/3,1/3,1/3]
S_3(high)		x	x		[0,1/2,1/2,0]

$$Entropy(S_1) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - 0 - 0 - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1.0$$

$$Entropy(S_2) = 0 - \frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 1.585$$

$$Entropy(S_3) = 0 - \frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) - 0 = 1.0$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid Fever) = \frac{2}{7}\cdot 1 + \frac{3}{7}\cdot 1.585 + \frac{2}{7}\cdot 1 = 1.2507$$

Vomiting attribute:

Values	H	I	S	B	Entropy(S_i)
S_1 (yes)			x	xx	[0,0,1/3, 2/3] = 0.918
S_2(no)	x	xx	x		[1/4, 2/4, 1/4, 0] = 1.5

Therefore, the entropy for this attribute is:

$$Entropy(S \mid Vomiting) = \frac{3}{7} \cdot 0.918 + \frac{4}{7} \cdot 1.5 = 1.2505$$

Diarrhea attribute:

Values	H	I	S	B	Entropy(S_i)
S_1 (yes)			xx	xx	[0,0,2/4, 2/4] = 1.0
S_2(no)	x	xx			[1/3, 2/3, 0, 0] = 0.918

Therefore, the entropy for this attribute is:

$$Entropy(S \mid Diarrhea) = \frac{4}{7} \cdot 1.0 + \frac{3}{7} \cdot 0.918 = 0.965$$

Shivering attribute:

Values	H	I	S	B	Entropy(S_i)
S_1 (yes)		x			[0,1,0,0] = 0
S_2(no)	x	x	xx	xx	[1/6, 1/6, 2/6, 2/6] = 1.918

Therefore, the entropy for this attribute is:

$$Entropy(S \mid Shivering) = \frac{1}{7} \cdot 0 + \frac{6}{7} \cdot 1.918 = 1.644$$

So, we choose the attribute that maximizes the overall information gain

- Fever: Gain(S) = Ent(S) − Ent(S|Fever) = 1.948 − 1.2507 = 0.6973
- Vomiting: Gain(S) = Ent(S) − Ent(S|Vomit) = 1.948 − 1.2505 = 0.6975
- Diarrhea: Gain(S) = Ent(S) − Ent(S|Diarrh) = 1.948 − 0.965 = 0.983
- Shivering: Gain(S) = Ent(S) − Ent(S|Shiver) = 1.948 − 1.644 = 0.304

Thus, "Diarrhea" is the most effective one that maximizes the information gain.

We would like to predict the gender of a person based on two binary attributes: leg-cover (pants or skirts) and beard (beard or bare-faced). We assume we have a dataset of 20000 individuals, 12000 of which are male and 8000 of which are female. 80% of the 12000 males are barefaced. Skirts are present on 50% of the females. All females are bare-faced and no male wears a skirt. Calculate which attribute needs to be selected for the first (root) node. Justify your response.

For the calculation of $\log_2$, remember, from maths, that: $\log_a b = \dfrac{\log_{10} b}{\log_{10} a}$

Solution

We have 12000 males and 8000 females. Thus, the analogy for males is 12000/20000 = 3/5, while for females 2/5.

The parent entropy (i.e. total) is calculated as:

$$Entropy(S) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.971$$

We need to check the entropy for each attribute individually and then find which attribute maximizes the information gain.

Bare-faced attribute:

Values	Male	Female	Entropy(S_i)
S_1 (yes)	9600	8000	[9600/17600, 8000/17600]
S_2(no)	2400	0	[2400/2400, 0] = [1, 0]

$$Entropy(S_1) = -\frac{96}{176}\log_2\left(\frac{96}{176}\right) - \frac{80}{176}\log_2\left(\frac{80}{176}\right) = 0.994$$

$$Entropy(S_2) = -\log_2(1) - 0 = 0$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid bare-faced) = \frac{17600}{20000}\cdot 0.994 + \frac{2400}{20000}\cdot 0 = 0.8747$$

Leg-cover attribute:

Values	Male	Female	Entropy(S_i)
S_1 (yes)	0	4000	[0, 1]
S_2(no)	12000	4000	[12000/16000, 4000/16000]

$$Entropy(S_1) = 0 - \log_2(1) = 0$$

$$Entropy(S_2) = -\frac{12}{16}\log_2\left(\frac{12}{16}\right) - \frac{4}{16}\log_2\left(\frac{4}{16}\right) = 0.8112$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid leg-\mathrm{cov}er) = \frac{4000}{20000}\cdot 0 + \frac{16000}{20000}\cdot 0.8112 = 0.649$$

So, we choose the attribute that maximizes the overall information gain

- Bare-faced: Gain(S) = Ent(S) − Ent(S|bare-faced) = 0.971 – 0.8747 = 0.0963
- Leg-cover: Gain(S) = Ent(S) − Ent(S|leg-cover) = 0.971-0.649 = 0.322

Thus, "Leg-cover" is the most effective one that maximizes the information gain.

We would like to indicate what factor(s) may affect sunburn. We have the following dataset of 8 samples, with attributes like Hair, Height, Weight and Lotion. The question is practically to create a decision tree, based on this dataset, and try to construct the tree structure. Do we need all these attributes?

Name	**Hair**	**Height**	**Weight**	**Lotion**	**Result**
Sarah	blonde	average	light	no	sunburned (positive)
Dana	blonde	tall	average	yes	none (negative)
Alex	brown	short	average	yes	none
Annie	blonde	short	average	no	sunburned
Emily	red	average	heavy	no	sunburned
Pete	brown	tall	heavy	no	none
John	brown	average	heavy	no	none
Katie	blonde	short	light	yes	none

Solution

We have 3+ (positives) and 5- (negatives) cases, from the result column.
The parent entropy (i.e. total) is calculated as:

$$Entropy(S) = -\frac{3}{8}\log_2\left(\frac{3}{8}\right) - \frac{5}{8}\log_2\left(\frac{5}{8}\right) = 0.9544$$

We need to check the entropy for each attribute individually and then find which attribute maximizes the information gain (in order to be used at root position).

Hair attribute:

Values	positive	negative	Entropy(S_i)
S_{Blonde}	2	2	[2/4, 2/4] = [1/2, 1/2]
S_{Brown}	0	3	[0, 3/3] = [0, 1]
S_{Red}	1	0	[1, 0]

$$Entropy(S_{Blonde}) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$Entropy(S_{brown}) = 0 - \log_2(1) = 0$$

$$Entropy(S_{Red}) = -\log_2(1) - 0 = 0$$

Therefore, the entropy for this attribute is:

$$Entropy(S \mid hair) = \frac{4}{8} \cdot 1.0 + \frac{3}{8} \cdot 0 + \frac{1}{8} \cdot 0 = 0.5$$

The information gain for Hair is: Entropy(S) – Entropy(S | hair) = 0.9544-0.5=0.4544

Height attribute:

Values	positive	negative	Entropy(S_i)
$S_{average}$	2	1	[2/3, 1/3]
S_{tall}	0	2	[0, 2/2] = [0, 1]
S_{short}	1	2	[1/3, 2/3]

Entropy ($S_{Average}$) = 0.91829 check these results
Entropy (S_{Tall}) = 0
Entropy (S_{Short}) = 0.91829

Therefore, the entropy for this attribute is:
Entropy(S | Height) = (3/8)*0.91829 + (2/8)*0 + (3/8)*0.91829 = 0.6887
The information gain for Height is: Entropy(S) – Entropy(S | height) = 0.9544-0.6887=0.2657

Weight attribute:

Values	positive	negative	Entropy(S_i)
S_{light}	1	1	[1/2, 1/2]
$S_{average}$	1	2	[1/3, 2/3]
S_{heavy}	1	2	[1/3, 2/3]

Entropy (S_{light}) = 1 check these results
Entropy ($S_{average}$) = 0.91829
Entropy (S_{heavy}) = 0.91829

Therefore, the entropy for this attribute is:
Entropy(S | Weight) = (2/8)*1 + (3/8)*0.91829 + (3/8)*0.91829 = 0.9387
The information gain for Weight is: Entropy(S) – Entropy(S | weight) = 0.9544-0.9387=0.0157

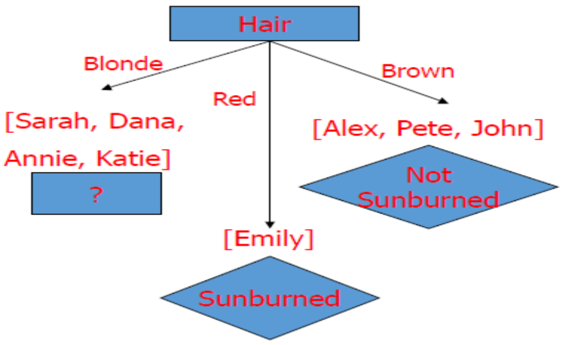
Lotion attribute:

Values	positive	negative	Entropy(S_i)
S_{yes}	0	3	[0, 1]
S_{no}	3	2	[3/5, 2/5]

Entropy (S_{yes}) = 0 check these results
Entropy (S_{no}) = 0.97095

Therefore, the entropy for this attribute is:
Entropy(S | Lotion) = (3/8)*0 + (5/8)*0.97095 = 0.6068
The information gain for Lotion is: Entropy(S) – Entropy(S | lotion) = 0.9544-0.6068=0.3475

Thus, based on the information gain for each attribute, Hair is the one chosen for the root node. But we need to proceed to the creation of the remaining decision tree.



As we can see, Hair seems to be a good choice, solved the tree regarding brown and red hairs, but not for blonde. So for this, we need another attribute to look for. We need to concentrate only for the "blonde" samples.

Name	Hair	Height	Weight	Lotion	Sunburned
Sarah	Blonde	Average	Light	No	Yes
Dana	Blonde	Tall	Average	Yes	No
Annie	Blonde	Short	Average	No	Yes
Katie	Blonde	Short	Light	Yes	No

Practically, we repeat the same procedure, but only for this limited number of samples.

We have 2+ (positives) and 2- (negatives) cases, from the result column.
The parent entropy (i.e. total) is calculated as:

$$Entropy(S) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

We need to check the entropy for each attribute individually (except hair) and then find which attribute maximizes the information gain (in order to be used at that missing position).

Height attribute:

Values	positive	negative	Entropy(S_i)
$S_{average}$	1	0	[1, 0]
S_{tall}	0	1	[0, 1]
S_{short}	1	1	[1/2, 1/2]

Entropy ($S_{Average}$) = 0   check these results
Entropy (S_{Tall}) = 0
Entropy (S_{Short}) = 1

Therefore, the entropy for this attribute is:
Entropy(S | Height) = (1/4)*0 + (1/4)*0 + (2/4)*1 = 0.5
The information gain for Height is: Entropy(S) – Entropy(S | height) = 1.0-0.5=0.5

Weight attribute:

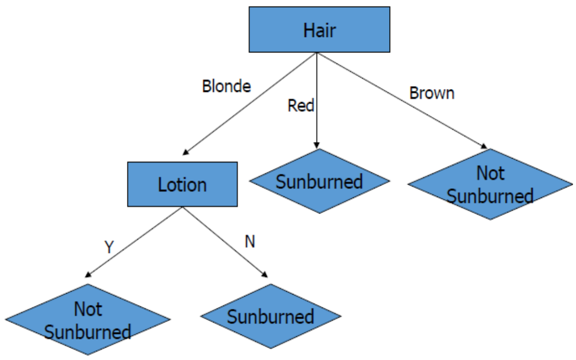
$S_{Average}$ = [1+, 1-]   E($S_{Average}$) = 1
S_{Light} = [1+, 1-]   E(S_{Light}) = 1
Information Gain(S, Weight) = 1 – [(2/4)*1 + (2/4)*1] = 0

Lotion attribute:

S_{Yes} = [0+, 2-] E(S_{Yes}) = 0
S_{No} = [2+, 0-] E(S_{No}) = 0

Information Gain(S, Lotion) = 1 – [(2/4)*0 + (2/4)*0] = 1

So, Lotion is the chosen one.



Consider the following dataset of houses represented by 5 training examples. The target attribute is 'Acceptable', which can have values 'yes' or 'no'. This is to be predicted based on the other attributes of the house.

House	Furniture	Nr rooms	New kitchen	Acceptable
1	No	3	Yes	Yes
2	Yes	3	No	No
3	No	4	No	Yes
4	No	3	No	No
5	Yes	4	No	Yes

Calculate the entropy of the target attribute. Construct the decision tree from the above dataset, using the information gain criterion as a measurement for the split decision.

This is the total entropy. We have 3+ (yes) and 2- (no) cases, from the acceptable column. The parent entropy (i.e. total) is calculated as:

$$Entropy(S) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971$$

The question is which one of these three attributes can be considered as root node.

<u>Furniture attribute:</u>

Values	yes	no	Entropy(S_i)
S_{yes}	1	1	[1/2, 1/2]
S_{no}	2	1	[2/3, 1/3]

$$Entropy(S_{yes}) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$Entropy(S_{no}) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 0.91829$$

Therefore, the entropy for this attribute is:
Entropy(S | furniture) = (2/5)*1 + (3/5)*0.91829 = 0.9509
Information Gain for this attribute: 0.971-0.9509 = 0.0202

Do the same for the other two attributes:
Gain(S |Nr Rooms) =0.971-0.5508=0.4202 - winner
Gain(S |New Kitchen) =0.971-0.8=0.171

Continue with the remaining attributes:
Step 2:
Entropy (S) = 0.918
Gain(S,Furniture) = 0.918-2/3 = 0.2513
Gain(S,New Kitchen) = 0.918-0=0.918 winner