

# Q1(a)

a) What is the entropy of this dataset with respect to the target class label *Result*?

	Name	Hair	Height	Build	Lotion	Result
1	Sarah	blonde	average	light	no	sunburned
2	Dana	blonde	tall	average	yes	none
3	Alex	brown	short	average	yes	none
4	Annie	blonde	short	average	no	sunburned
5	Emily	red	average	heavy	no	sunburned
6	Pete	brown	tall	heavy	no	none
7	John	brown	average	heavy	no	none
8	Katie	brown	short	light	yes	none

2 classes:  $p_1=(3/8)$   $p_2=(5/8)$

Entropy(Dataset)

$= -(3/8) \times \log_2(3/8) - (5/8) \times \log_2(5/8)$

$= 0.5306 + 0.4238 = 0.9544$

$$H(S) = - \sum_{i=1}^n p_i \log_2 p_i$$

Assume  $\log_2(0)=0$

Also  $\log_2(x)=\log_{10}(x)/\log_{10}(2)$

## Q1(b)

---

b) Construct the decision tree that would be built with Information Gain for this data set. Show your work for selection of the root feature in your tree.

	Name	Hair	Height	Build	Lotion	Result
1	Sarah	blonde	average	light	no	sunburned
2	Dana	blonde	tall	average	yes	none
3	Alex	brown	short	average	yes	none
4	Annie	blonde	short	average	no	sunburned
5	Emily	red	average	heavy	no	sunburned
6	Pete	brown	tall	heavy	no	none
7	John	brown	average	heavy	no	none
8	Katie	brown	short	light	yes	none

- Steps:**
1. Calculate overall dataset entropy (Done).
  2. Calculate entropy for each feature.
  3. Calculate Information Gain for each feature.

# Q1(b)

---

Calculate entropy values for all features by looking at number of times each possible value occurs at root node:

Entropy(Hair=blonde) =

Entropy(Hair=brown) =

Entropy(Hair=red) =

Hair	Result
blonde	sunburned
blonde	none
blonde	sunburned
brown	none
brown	none
brown	none
brown	none
red	sunburned

# Q1(b)

Calculate entropy values for all features by looking at number of times each possible value occurs at root node:

$$\text{Entropy}(\text{Hair}=\text{blonde}) = \text{Entropy}(1/3, 2/3) = 0.9183$$

$$\text{Entropy}(\text{Hair}=\text{brown}) = \text{Entropy}(0/4, 4/4) = 0$$

$$\text{Entropy}(\text{Hair}=\text{red}) = \text{Entropy}(1/1, 0/1) = 0$$

$$\text{Entropy}(\text{Height}=\text{average}) = \text{Entropy}(1/3, 2/3) = 0.9183$$

$$\text{Entropy}(\text{Height}=\text{tall}) = \text{Entropy}(2/2, 0/2) = 0$$

$$\text{Entropy}(\text{Height}=\text{short}) = \text{Entropy}(1/3, 2/3) = 0.9183$$

Similarly...

$$\text{Entropy}(\text{Build}=\text{light}) = \text{Entropy}(1/2, 1/2) = 1$$

$$\text{Entropy}(\text{Build}=\text{average}) = \text{Entropy}(1/3, 2/3) = 0.9183$$

$$\text{Entropy}(\text{Build}=\text{heavy}) = \text{Entropy}(1/3, 2/3) = 0.9183$$

$$\text{Entropy}(\text{Lotion}=\text{no}) = \text{Entropy}(2/5, 3/5) = 0.9710$$

$$\text{Entropy}(\text{Lotion}=\text{yes}) = \text{Entropy}(3/3, 0/3) = 0$$

Hair	Result
blonde	sunburned
blonde	none
blonde	sunburned
brown	none
brown	none
brown	none
brown	none
red	sunburned

Height	Result
average	sunburned
average	sunburned
average	none
short	none
short	sunburned
short	none
tall	none
tall	none

# Q1(b)

Use Information Gain to choose best feature to split for root node. Try each feature in turn...

$$\begin{aligned} \text{IG}(\text{Hair}) &= \text{Entropy}(\text{Dataset}) \\ &- p(\text{Hair}=\text{blonde}) \times \text{Entropy}(\text{Hair}=\text{blonde}) \\ &- p(\text{Hair}=\text{brown}) \times \text{Entropy}(\text{Hair}=\text{brown}) \\ &- p(\text{Hair}=\text{red}) \times \text{Entropy}(\text{Hair}=\text{red}) \\ &= 0.9544 - (3/8) \times 0.9183 - (4/8) \times 0 - (1/8) \times 0 \\ &= 0.610 \end{aligned}$$

$$\text{Entropy}(\text{Dataset}) = 0.9544$$

$$\text{Entropy}(\text{Hair}=\text{blonde}) = 0.9183$$

$$\text{Entropy}(\text{Hair}=\text{brown}) = 0$$

$$\text{Entropy}(\text{Hair}=\text{red}) = 0$$

$$\text{Entropy}(\text{Height}=\text{average}) = 0.9183$$

$$\text{Entropy}(\text{Height}=\text{tall}) = 0$$

$$\text{Entropy}(\text{Height}=\text{short}) = 0.9183$$

$$\text{Entropy}(\text{Build}=\text{light}) = 1$$

$$\text{Entropy}(\text{Build}=\text{average}) = 0.9183$$

$$\text{Entropy}(\text{Build}=\text{heavy}) = 0.9183$$

$$\text{Entropy}(\text{Lotion}=\text{no}) = 0.9710$$

$$\text{Entropy}(\text{Lotion}=\text{yes}) = 0$$

$$\text{IG}(\text{Height}) = 0.9544 - (3/8) \times 0.9183 - (2/8) \times 0 - (3/8) \times 0.9183 = 0.2657$$

$$\text{IG}(\text{Build}) = 0.9544 - (2/8) \times 1 - (3/8) \times 0.9183 - (3/8) \times 0.9183 = 0.0157$$

$$\text{IG}(\text{Lotion}) = 0.9544 - (5/8) \times 0.9710 - (3/8) \times 0 = 0.3475$$

➡ “Hair” will be selected as the feature with the highest IG value.  
It perfectly classifies the data for *Hair=brown* & *Hair=red*

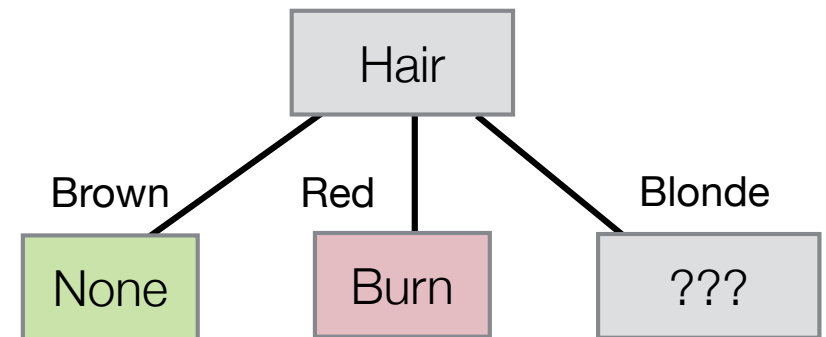
# Q1(b)

---

- “Hair” selected as the feature with the highest IG value  $\Rightarrow$  used to split the root node of the tree.

Child node *Hair=blonde*:

	Name	Hair	Height	Build	Lotion	Result
1	Sarah	blonde	average	light	no	sunburned
2	Dana	blonde	tall	average	yes	none
4	Annie	blonde	short	average	no	sunburned

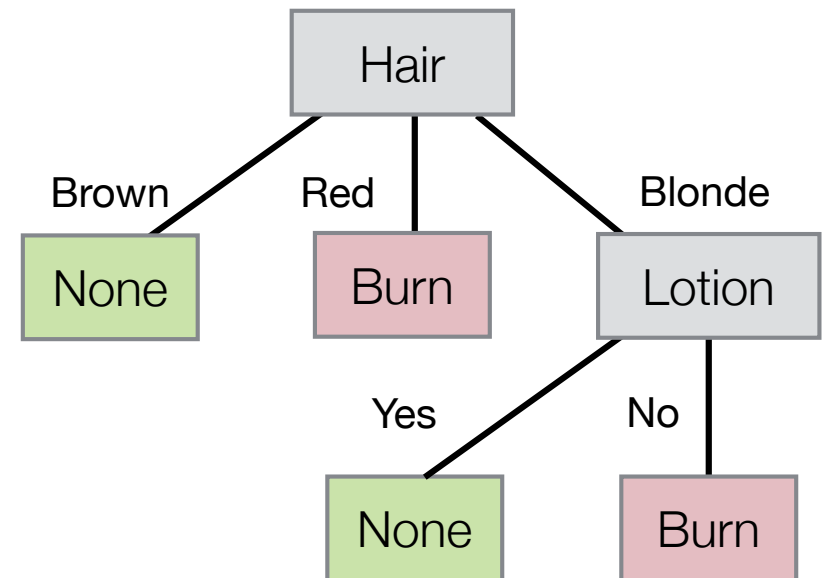


## Q1(b)

- “Hair” selected as the feature with the highest IG value  $\Rightarrow$  used to split the root node of the tree.

Child node *Hair=blonde*:

	Name	Hair	Height	Build	Lotion	Result
1	Sarah	blonde	average	light	no	sunburned
2	Dana	blonde	tall	average	yes	none
4	Annie	blonde	short	average	no	sunburned



- ➡ The case for *Hair=blonde* contains (2 sunburned, 1 none).  
Can split these into pure child nodes using feature “Lotion”.

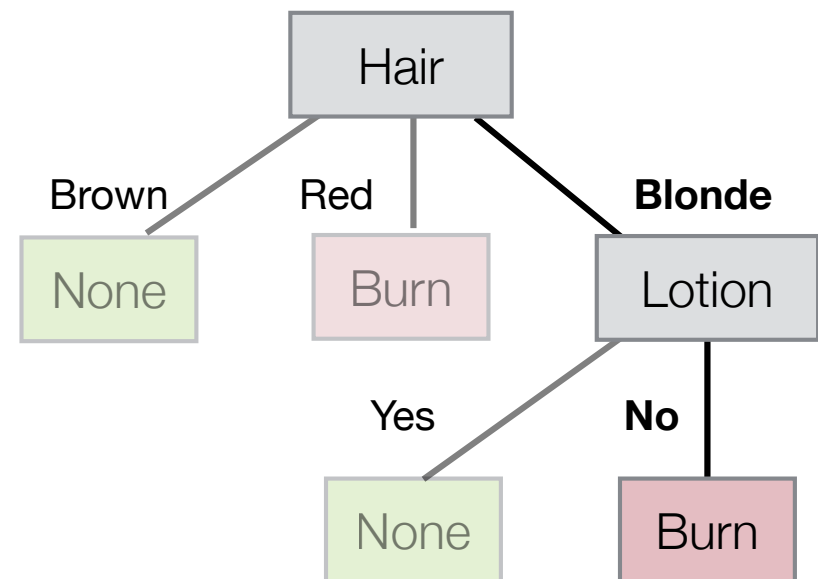
## Q1(c)

---

c) Using your decision tree from (b), how would you classify the following example?

	Hair	Height	Build	Lotion	Result
<b>X</b>	blonde	average	heavy	no	???

- First, check *Hair=Blonde*
- Next, check *Lotion=No*
- **Output: Sunburned**





## Q2(a)

a) What is the entropy of this dataset with respect to the target class label Result based on the 14 examples above?

Example	Credit_History	Debt	Income	Risk
1	bad	low	0to30	high
2	bad	high	30to60	high
3	bad	low	0to30	high
4	unknown	high	30to60	high
5	unknown	high	0to30	high
6	good	high	0to30	high
7	bad	low	over60	medium
8	unknown	low	30to60	medium
9	good	high	30to60	medium
10	unknown	low	over60	low
11	unknown	low	over60	low
12	good	low	over60	low
13	good	high	over60	low
14	good	high	over60	low

$$H(S) = - \sum_{i=1}^n p_i \log_2 p_i$$

NB: Define  $\log_2(0)=0$

$$p1=(6/14)$$

$$p2=(3/14)$$

$$p3=(5/14)$$

$$\begin{aligned} & -(6/14) \times \log_2(6/14) - (3/14) \times \log_2(3/14) - (5/14) \times \log_2(5/14) \\ & = 0.5239 + 0.4762 + 0.5305 \\ & = 1.5306 \end{aligned}$$

## Q2(b)

b) Compute the entropy of each of the 3 descriptive features.

$$\text{Entropy}(\text{CH}=\text{bad}) = -(1/4) \times \log_2(1/4) - (3/4) \times \log_2(3/4) = 0.8113$$

$$\text{Entropy}(\text{CH}=\text{unknown}) = -(2/5) \times \log_2(2/5) - (1/5) \times \log_2(1/5) - (2/5) \times \log_2(2/5) = 1.5219$$

$$\text{Entropy}(\text{CH}=\text{good}) = -(1/5) \times \log_2(1/5) - (1/5) \times \log_2(1/5) - (3/5) \times \log_2(3/5) = 1.3710$$

	CH	Debt	Income	Risk
1	bad	low	0to30	high
2	bad	high	30to60	high
3	bad	low	0to30	high
4	unknown	high	30to60	high
5	unknown	high	0to30	high
6	good	high	0to30	high
7	bad	low	over60	medium
8	unknown	low	30to60	medium
9	good	high	30to60	medium
10	unknown	low	over60	low
11	unknown	low	over60	low
12	good	low	over60	low
13	good	high	over60	low
14	good	high	over60	low

$$\text{Entropy}(\text{Debt}=\text{low}) = -(2/7) \times \log_2(2/7) - (2/7) \times \log_2(2/7) - (3/7) \times \log_2(3/7) = 1.5567$$

$$\text{Entropy}(\text{Debt}=\text{high}) = -(4/7) \times \log_2(4/7) - (1/7) \times \log_2(1/7) - (2/7) \times \log_2(2/7) = 1.3788$$

$$\text{Entropy}(\text{Income}=\text{0to30}) = -(4/4) \times \log_2(4/4) = 0$$

$$\text{Entropy}(\text{Income}=\text{30to60}) = -(2/4) \times \log_2(2/4) - (2/4) \times \log_2(2/4) = 1$$

$$\text{Entropy}(\text{Income}=\text{over60}) = -(1/6) \times \log_2(1/6) - (5/6) \times \log_2(5/6) = 0.65$$

## Q2(c)

---

c) Which one of the predicting features would be selected by ID3 at the root of a decision tree? Explain your answer.

Use Information Gain to choose best feature to split for root node...

$$\begin{aligned} \text{IG}(\text{CH}) &= \text{Entropy}(\text{Dataset}) \\ &\quad - p(\text{CH}=\text{bad}) \times \text{Entropy}(\text{CH}=\text{bad}) \\ &\quad - p(\text{CH}=\text{unknown}) \times \text{Entropy}(\text{CH}=\text{unknown}) \\ &\quad - p(\text{CH}=\text{good}) \times \text{Entropy}(\text{CH}=\text{good}) \\ &= 1.5306 - (4/14) \times 0.8113 - (5/14) \times 1.5219 \\ &\quad - (5/14) \times 1.3710 \\ &= 0.2656 \end{aligned}$$

$\text{Entropy}(\text{CH}=\text{bad}) = 0.8113$
$\text{Entropy}(\text{CH}=\text{unknown}) = 1.5219$
$\text{Entropy}(\text{CH}=\text{good}) = 1.3710$
$\text{Entropy}(\text{Debt}=\text{low}) = 1.5567$
$\text{Entropy}(\text{Debt}=\text{high}) = 1.3788$
$\text{Entropy}(\text{Income}=\text{0to30}) = 0$
$\text{Entropy}(\text{Income}=\text{30to60}) = 1$
$\text{Entropy}(\text{Income}=\text{over60}) = 0.65$

$$\text{IG}(\text{Debt}) = 1.5306 - (7/14) \times 1.5567 - (7/14) \times 1.3788 = 0.0628$$

$$\text{IG}(\text{Income}) = 1.5306 - (4/14) \times 0 - (4/14) \times 1 - (6/14) \times 0.65 = 0.966$$

➡ “Income” will be selected as the feature to split as it has the highest IG value.

## Q3

---

- For the datasets analysed in the 03 **DTrees** notebook, will the resulting trees be different if the feature selection criterion is 'gini' instead of 'entropy'.

```
tree = DecisionTreeClassifier(criterion='gini')  
ap_tree = tree.fit(X, y)
```

# Q4

---

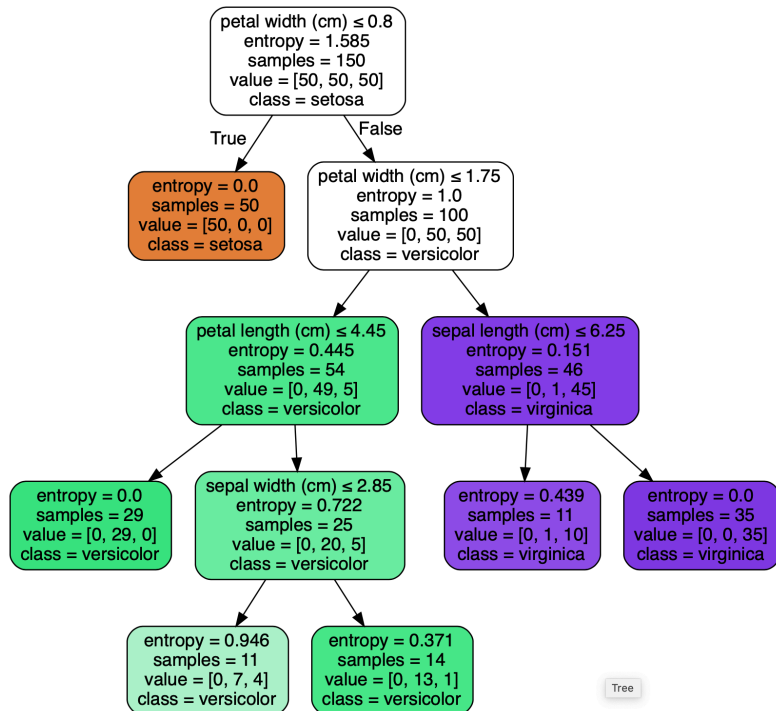
If a decision tree is allowed to be too *bushy* it is likely to overfit the training data. Consequently decision trees are often pruned to prevent overfitting.

In the example in the ``03 DTrees Lab`` notebook we use the `min_samples_leaf` attribute to control the size of the tree.

1. What does the Iris Data tree look like when no pruning is enforced?
2. What other options does sklearn provide to manage the bushiness of the tree? <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
3. Use two other pruning strategies to produce similar trees.

# Q4

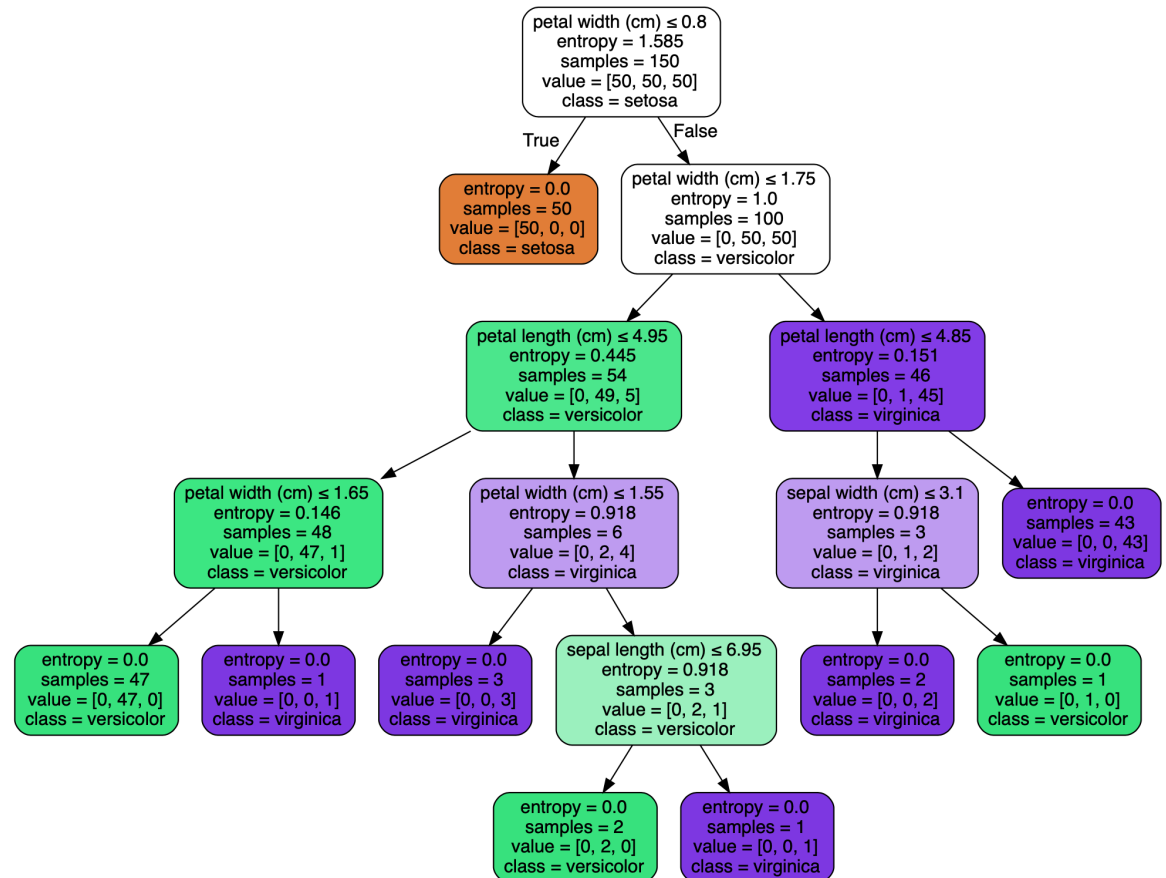
## 1. What does the Iris Data tree look like when no pruning is enforced?



```
i_tree.get_n_leaves()
```

```
Out[17]:
```

```
6
```



```
tree = DecisionTreeClassifier(criterion='entropy',
                              #min_samples_leaf=10
                              )
```

# Q4

---

2. What other options does sklearn provide to manage the bushiness of the tree? <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

**max\_depth** *int, default=None*

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.

**min\_samples\_split** *int or float, default=2*

The minimum number of samples required to split an internal node:

**min\_samples\_leaf** *int or float, default=1*

The minimum number of samples required to be at a leaf node.

**max\_leaf\_nodes** *int, default=None*

Grow a tree with max\_leaf\_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity.

**min\_impurity\_decrease** *float, default=0.0*

A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

**min\_impurity\_split** *float, default=0*

Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it is a leaf.