Solutions for Tutorial exercises
Backpropagation neural networks, Naïve Bayes, Decision Trees, k-NN, Associative Classification.

**Exercise 1.**
Suppose we want to classify potential bank customers as good creditors or bad creditors for loan applications. We have a training dataset describing past customers using the following attributes: Marital status \{married, single, divorced\}, Gender \{male, female\}, Age \{[18..30[, [30..50[, [50..65[, [65+]\}, Income \{[10K..25K[, [25K..50K[, [50K..65K[, [65K..100K[, [100K+]\}.

Design a neural network that could be trained to predict the credit rating of an applicant.

**Solution:**
We have 2 classes, good creditor and bad creditor. This means we would need two nodes in the output layer. There are 4 variables: Marital Status, Gender, Age and Income. However, since we have 3 values for Marital status, 2 values for Gender, 4 intervals for Age and 5 intervals for Income, we would have 14 neuron units in the input layer. In the hidden layer we can have \((14+2)/2=8\) neurons.

The architecture of the neural networks could look like this.

The weights are initialized with random values.

Are there other possible architectures?
Exercise 2.
Given the following neural network with initialized weights as in the picture, explain the network architecture knowing that we are trying to distinguish between nails and screws and an example of training tuples is as follows: T1{0.6, 0.1, nail}, T2{0.2, 0.3, screw}.

Let the learning rate \( \eta \) be 0.1 and the weights be as indicated in the figure above. Do the forward propagation of the signals in the network using T1 as input, then perform the back propagation of the error. Show the changes of the weights.

Solution:

What encoding of the outputs?
10 for class “nail”, 01 for class “screw”

Forward pass for T1 - calculate the outputs o6 and o7

\[
\begin{align*}
o1 &= 0.6, \quad o2 = 0.1, \quad \text{target output} \quad 10, \quad \text{i.e. class “nail”} \\
\text{Activations of the hidden units:} \\
\text{net3} &= o1 \times w13 + o2 \times w23 + b3 = 0.6 \times 0.1 + 0.1 \times (-0.2) + 0.1 = 0.14 \\
o3 &= \frac{1}{1 + e^{-\text{net3}}} = 0.53 \\
\text{net4} &= o1 \times w14 + o2 \times w24 + b4 = 0.6 \times 0 + 0.1 \times 0.2 + 0.2 = 0.22 \\
o4 &= \frac{1}{1 + e^{-\text{net4}}} = 0.55 \\
\text{net5} &= o1 \times w15 + o2 \times w25 + b5 = 0.6 \times 0.3 + 0.1 \times (-0.4) + 0.5 = 0.64 \\
o5 &= \frac{1}{1 + e^{-\text{net5}}} = 0.65 \\
\text{Activations of the output units:} \\
\text{net6} &= o3 \times w36 + o4 \times w46 + o5 \times w56 + b6 = 0.53 \times (-0.4) + 0.55 \times 0.1 + 0.65 \times 0.6 - 0.1 = 0.13 \\
o6 &= \frac{1}{1 + e^{-\text{net6}}} = 0.53 \\
\text{net7} &= o3 \times w37 + o4 \times w47 + o5 \times w57 + b7 = 0.53 \times 0.2 + 0.55 \times (-0.1) + 0.65 \times (-0.2) + 0.6 = 0.52 \\
o7 &= \frac{1}{1 + e^{-\text{net7}}} = 0.63
\end{align*}
\]

Backward pass for T1 - calculate the output errors \( \delta_6 \) and \( \delta_7 \)
(note that \( d6=1, \quad d7=0 \) for class “nail”)

\[ \delta_6 = (d6-o6) * o6 * (1-o6) = (1-0.53)*0.53*(1-0.53) = 0.12 \]
\[ \delta_7 = (d7-o7) * o7 * (1-o7) = (0-0.63)*0.63*(1-0.63) = -0.15 \]

**Calculate the new weights between the hidden and output units (\(\eta=0.1\))**

\[ \Delta w_{36} = \eta * \delta_6 * o3 = 0.1*0.12*0.53 = 0.006 \]
\[ w_{36\text{new}} = w_{36\text{old}} + \Delta w_{36} = 0.4+0.006 = 0.394 \]
\[ \Delta w_{37} = \eta * \delta_7 * o3 = 0.1*-0.15*0.53 = -0.008 \]
\[ w_{37\text{new}} = w_{37\text{old}} + \Delta w_{37} = 0.2-0.008 = 0.19 \]

Similarly for \(w_{46\text{new}}\), \(w_{47\text{new}}\), \(w_{56\text{new}}\) and \(w_{57\text{new}}\)

For the biases \(b_{6}\) and \(b_{7}\) (remember: biases are weights with input 1):
\[ \Delta b_6 = \eta * \delta_6 * 1 = 0.1*0.12 = 0.012 \]
\[ b_{6\text{new}} = b_{6\text{old}} + \Delta b_6 = -0.1+0.012 = -0.088 \]

Similarly for \(b_7\)

**Calculate the errors of the hidden units \(\delta_3\), \(\delta_4\) and \(\delta_5\)**

\[ \delta_3 = o3 * (1-o3) * (w_{36} * \delta_6 + w_{37} * \delta_7) = 0.53*(1-0.53)(-0.4*0.12+0.2*(-0.15)) = -0.019 \]

Similarly for \(\delta_4\) and \(\delta_5\)

**Calculate the new weights between the input and hidden units (\(\eta=0.1\))**

\[ \Delta w_{13} = \eta * \delta_3 * o1 = 0.1*(-0.019)*0.6 = -0.0011 \]
\[ w_{13\text{new}} = w_{13\text{old}} + \Delta w_{13} = 0.1-0.0011 = 0.0989 \]

Similarly for \(w_{23\text{new}}\), \(w_{14\text{new}}\), \(w_{24\text{new}}\), \(w_{15\text{new}}\) and \(w_{25\text{new}}\); \(b_3\), \(b_4\) and \(b_6\)

Repeat the same procedure for the other training examples

Forward pass for T2...backward pass for T2...

**Exercise 3.**

Why is the Naïve Bayesian classification called “naïve”?

**Answer:** Naïve Bayes assumes that all attributes are: 1) equally important and 2) independent of one another given the class.
**Exercise 4. Naïve Bayes for data with nominal attributes**

Given the training data in the table below (Buy Computer data), predict the class of the following new example using Naïve Bayes classification: age<=30, income=medium, student=yes, credit-rating=fair

<table>
<thead>
<tr>
<th>RID</th>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>Class: buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>31...40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>&lt;=30</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>31...40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>31...40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

**Solution:**

E = age<=30, income=medium, student=yes, credit-rating=fair
E1 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.

\[
P(\text{yes} \mid E) = \frac{P(E_1 \mid \text{yes}) P(E_2 \mid \text{yes}) P(E_4 \mid \text{yes}) P(\text{yes})}{P(E)}
\]

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

\[
P(\text{yes} \mid E) = \frac{0.222 \cdot 0.444 \cdot 0.667 \cdot 0.668 \cdot 0.443}{P(E)} = 0.028 \quad P(\text{no} \mid E) = \frac{0.6 \cdot 0.4 \cdot 0.2 \cdot 0.4 \cdot 0.357}{P(E)} = 0.007
\]

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

**Exercise 5. Applying Naïve Bayes to data with numerical attributes and using the Laplace correction (to be done at your own time, not in class)**

Given the training data in the table below (Tennis data with some numerical attributes), predict the class of the following new example using Naïve Bayes classification: outlook=overcast, temperature=60, humidity=62, windy=false.
Tip. You can use Excel or Matlab for the calculations of logarithm, mean and standard deviation. Matlab is installed on our undergraduate machines. The following Matlab functions can be used: \( \log_2 \) – logarithm with base 2, mean – mean value, std – standard deviation. Type \texttt{help<function name>} (e.g. \texttt{help mean}) for help on how to use the functions and examples.

<table>
<thead>
<tr>
<th>outlook</th>
<th>temperature</th>
<th>humidity</th>
<th>windy</th>
<th>play</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>85</td>
<td>85</td>
<td>false</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>80</td>
<td>90</td>
<td>true</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>83</td>
<td>66</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>70</td>
<td>96</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>68</td>
<td>80</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>65</td>
<td>70</td>
<td>true</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>64</td>
<td>65</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>72</td>
<td>95</td>
<td>false</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>69</td>
<td>70</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>75</td>
<td>80</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>75</td>
<td>70</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>72</td>
<td>90</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>81</td>
<td>75</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>71</td>
<td>91</td>
<td>true</td>
<td>no</td>
</tr>
</tbody>
</table>

\[ \mu = \frac{\sum_{i=1}^{n} X_i}{n} \]
\[ \sigma^2 = \frac{\sum_{i=1}^{n} (X_i - \mu)^2}{n-1} \]

\( \mu_{\text{temp\_yes}}=73, \sigma_{\text{temp\_yes}}=6.2; \quad \mu_{\text{temp\_no}}=74.6, \sigma_{\text{temp\_no}}=8.0 \)

\( \mu_{\text{hum\_yes}}=79.1, \sigma_{\text{temp\_yes}}=10.2; \quad \mu_{\text{hum\_no}}=86.2, \sigma_{\text{temp\_no}}=9.7 \)

Second, to calculate \( f(\text{temperature}=60|\text{yes}) \), \( f(\text{temperature}=60|\text{no}) \), \( f(\text{humidity}=62|\text{yes}) \) and \( f(\text{humidity}=62|\text{no}) \) using the probability density function for the normal distribution:

\[ f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
Third, we can calculate the probabilities for the nominal attributes:

\[ P(\text{yes}) = \frac{9}{14} = 0.643 \]
\[ P(\text{no}) = \frac{5}{14} = 0.357 \]
\[ P(\text{outlook=overcast}|\text{yes}) = \frac{4}{14} = 0.286 \]
\[ P(\text{outlook=overcast}|\text{no}) = \frac{0}{5} = 0 \]
\[ P(\text{windy=false}|\text{yes}) = \frac{6}{9} = 0.667 \]
\[ P(\text{windy=false}|\text{no}) = \frac{2}{5} = 0.4 \]

As \( P(\text{outlook=overcast}|\text{no}) = 0 \), we need to use a Laplace estimator for the attribute outlook. We assume that the three values (sunny, overcast, rainy) are equally probable and set \( \mu = 3 \):

\[ P(\text{outlook = overcast}|\text{yes}) = \frac{4 + 1}{9 + 3} = \frac{5}{12} = 0.4167 \]
\[ P(\text{outlook = overcast}|\text{no}) = \frac{0 + 1}{5 + 3} = \frac{1}{8} = 0.125 \]

Fourth, we can calculate the final probabilities:

\[ P(\text{yes}|E) = \frac{0.4167 * 0.0071 * 0.0096 * 0.667 * 0.643}{P(E)} = 1.22 * 10^{-5} \]
\[ P(\text{no}|E) = \frac{0.125 * 0.0094 * 0.0018 * 0.4 * 0.357}{P(E)} = 3.02 * 10^{-7} \]

Therefore, the Naïve Bayes classifier predicts play=yes for the new example.

---

**Exercise 6. Using Weka (to be done at your own time, not in class)**

Load iris data (iris.arff). Choose 10-fold cross validation. Run the Naïve Bayes and Multi-layer perceptron (trained with the backpropagation algorithm) classifiers and compare their performance. Which classifier produced the most accurate classification? Which one learns faster?

**Exercise 7. k-Nearest neighbours**

Given the training data in Exercise 4 (Buy Computer data), predict the class of the following new example using k-Nearest Neighbour for \( k=5 \): age\(\leq30 \), income=medium, student=yes, credit-rating=fair. For similarity measure use a simple match of attribute values: Similarity(A,B)=

\[ f(\text{temperature} = 60| \text{yes}) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(60-73)^2}{(60-74.6)^2}} = 0.071 \]
\[ f(\text{temperature} = 60| \text{no}) = \frac{1}{8\sqrt{2\pi}} e^{-\frac{8^2}{(60-74.6)^2}} = 0.0094 \]
\[ f(\text{humidity} = 62| \text{yes}) = \frac{1}{10.2\sqrt{2\pi}} e^{-\frac{(62-79.1)^2}{(62-86.2)^2}} = 0.0096 \]
\[ f(\text{humidity} = 62| \text{no}) = \frac{1}{9.7\sqrt{2\pi}} e^{-\frac{2(9.7)^2}{(62-86.2)^2}} = 0.0018 \]
\[
\sum_{i=1}^{4} w_i \cdot \partial(a_i, b_i) / 4
\] where \( \partial(a_i, b_i) \) is 1 if \( a_i \) equals \( b_i \) and 0 otherwise. \( a_i \) and \( b_i \) are either \text{age, income, student} or \text{credit_rating}. Weights are all 1 except for income it is 2.

**Solution:**

<table>
<thead>
<tr>
<th>RID</th>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>Class: buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>31...40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>&lt;=30</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>31...40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>31...40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RID</th>
<th>Class</th>
<th>Distance to New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No</td>
<td>(1+0+0+1)/4=0.5</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>(1+0+0+0)/4=0.25</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>(0+0+0+1)/4=0.25</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>(0+2+0+1)/4=0.75</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>(0+0+1+1)/4=0.5</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>(0+0+1+0)/4=0.25</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>(0+0+1+0)/4=0.25</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>(1+2+0+1)/4=1</td>
</tr>
<tr>
<td>9</td>
<td>Yes</td>
<td>(1+0+1+1)/4=0.75</td>
</tr>
<tr>
<td>10</td>
<td>Yes</td>
<td>(0+2+1+1)/4=1</td>
</tr>
<tr>
<td>11</td>
<td>Yes</td>
<td>(1+2+1+0)/4=1</td>
</tr>
<tr>
<td>12</td>
<td>Yes</td>
<td>(0+2+0+0)/4=0.5</td>
</tr>
<tr>
<td>13</td>
<td>Yes</td>
<td>(0+0+1+1)/4=0.5</td>
</tr>
<tr>
<td>14</td>
<td>No</td>
<td>(0+2+0+0)/4=0.5</td>
</tr>
</tbody>
</table>

Among the five nearest neighbours four are from class \text{Yes} and one from class \text{No}. Hence, the k-NN classifier predicts \text{buys_computer=yes} for the new example.

**Exercise 8. Decision trees**

Given the training data in Exercise 4 (\text{Buy Computer} data), build a decision tree and predict the class of the following new example: \text{age<=30, income=medium, student=yes, credit-rating=fair}.

**Solution:**

First check which attribute provides the highest Information Gain in order to split the training set based on that attribute. We need to calculate the expected information to classify the set and the entropy of each attribute. The information gain is this mutual information minus the entropy:
The mutual information of the two classes $I(S_{Yes}, S_{No}) = I(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$

- For Age we have three values $age_{<=30}$ (2 yes and 3 no), $age_{31..40}$ (4 yes and 0 no) and $age_{>40}$ (3 yes 2 no)

Entropy(age) = $5/14 (-2/5 \log(2/5)-3/5 \log(3/5)) + 4/14 (0) + 5/14 (-5/14 \log(5/14))$

= $5/14(0.9709) + 0 + 5/14(0.9709) = 0.6935$

Gain(age) = 0.94 – 0.6935 = 0.2465

- For Income we have three values $income_{high}$ (2 yes and 2 no), $income_{medium}$ (4 yes and 2 no) and $income_{low}$ (3 yes 1 no)

Entropy(income) = $4/14(-2/4 \log(2/4)-2/4 \log(2/4)) + 6/14 (-4/6 \log(4/6)-2/6 \log(2/6))$

= $4/14 (1) + 6/14 (0.918) + 4/14 (0.811) = 0.285714 + 0.393428 + 0.231714 = 0.9108$

Gain(income) = 0.94 – 0.9108 = 0.0292

- For Student we have two values $student_{yes}$ (6 yes and 1 no) and $student_{no}$ (3 yes 4 no)

Entropy(student) = $7/14(-6/7 \log(6/7)) + 7/14(-3/7 \log(3/7)-4/7 \log(4/7))$

= $7/14(0.5916) + 7/14(0.9852) = 0.2958 + 0.4926 = 0.7884$

Gain (student) = 0.94 – 0.7884 = 0.1516

- For Credit_Rating we have two values $credit\_rating_{fair}$ (6 yes and 2 no) and $credit\_rating_{excellent}$ (3 yes 3 no)

Entropy(credit_rating) = $8/14(-6/8 \log(6/8)-2/8 \log(2/8)) + 6/14(-3/6 \log(3/6)-3/6 \log(3/6))$

= $8/14(0.8112) + 6/14(1) = 0.4635 + 0.4285 = 0.8920$

Gain(credit_rating) = 0.94 – 0.8920 = 0.479

Since Age has the highest Information Gain we start splitting the dataset using the age attribute

Since all records under the branch age_{31..40} are all of class Yes, we can replace the leaf with Class=Yes.
The same process of splitting has to happen for the two remaining branches. 
For branch age <= 30 we still have attributes income, student and credit_rating. Which one should be use to split the partition?

The mutual information is \( I(S_{Yes}, S_{No}) = I(2,3) = \frac{-2}{5} \log_2(2/5) - \frac{3}{5} \log_2(3/5) = 0.97 \)

- For Income we have three values income_{high} (0 yes and 2 no), income_{medium} (1 yes and 1 no) and income_{low} (1 yes and 0 no)

  Entropy(income) = \( \frac{2}{5}(0) + \frac{2}{5}(-1/2\log(1/2)-1/2\log(1/2)) + 1/5 (0) \)
  = \( \frac{2}{5}(1) = 0.4 \)

  Gain(income) = 0.97 – 0.4 = 0.57

- For Student we have two values student_{yes} (2 yes and 0 no) and student_{no} (0 yes 3 no)

  Entropy(student) = \( \frac{2}{5}(0) + \frac{3}{5}(0) = 0 \)

  Gain (student) = 0.97 – 0 = 0.97

We can then safely split on attribute student without checking the other attributes since the information gain is maximized.

Since these two new branches are from distinct classes, we make them into leaf nodes with their respective class as label:
Again the same process is needed for the other branch of age.

The mutual information is $I(S_{\text{Yes}}, S_{\text{No}}) = I(3, 2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$

- For Income we have two values income$_{\text{medium}}$ (2 yes and 1 no) and income$_{\text{low}}$ (1 yes and 1 no)

$\text{Entropy(income)} = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5 (-1/2\log(1/2)-1/2\log(1/2))$

$= 3/5(0.9182)+2/5 (1) = 0.55+0. 4= 0.95$

$\text{Gain(income)} = 0.97 – 0.95 = 0.02$

- For Student we have two values student$_{\text{yes}}$ (2 yes and 1 no) and student$_{\text{no}}$ (1 yes and 1 no)

$\text{Entropy(student)} = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5(-1/2\log(1/2)-1/2\log(1/2)) = 0.95$

$\text{Gain (student)} = 0.97 – 0.95 = 0.02$

- For Credit_Rating we have two values credit_rating$_{\text{fair}}$ (3 yes and 0 no) and credit_rating$_{\text{excellent}}$ (0 yes and 2 no)

$\text{Entropy(credit_rating)} = 0$

$\text{Gain(credit_rating)} = 0.97 – 0 = 0.97$

We then split based on credit_rating. These splits give partitions each with records from the same class. We just need to make these into leaf nodes with their class label attached:

New example: age$\leq$30, income=medium, student=yes, credit-rating=fair
Follow branch(age$\leq$30) then student=yes we predict Class=yes \[ \Rightarrow \] Buys_computer = yes
Exercise 9. Associative Classifier

Given the training data in Exercise 4 (Buy Computer data), build an associative classifier model by generating all relevant association rules with support and confidence thresholds 10% and 60% respectively. Classify using this model the new example: age<=30, income=medium, student=yes, credit-rating=fair, selecting the rule with the highest confidence. What would be the classification if we chose to vote the class among all rules that apply?

Solution:
If we map the attribute values as follows the training set becomes the following transactions:

<table>
<thead>
<tr>
<th>Attribute value</th>
<th>New symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>age &lt;= 30</td>
<td>a</td>
</tr>
<tr>
<td>age &gt; 30</td>
<td>c</td>
</tr>
<tr>
<td>income high</td>
<td>h</td>
</tr>
<tr>
<td>income medium</td>
<td>m</td>
</tr>
<tr>
<td>income low</td>
<td>l</td>
</tr>
<tr>
<td>student yes</td>
<td>s</td>
</tr>
<tr>
<td>student no</td>
<td>t</td>
</tr>
<tr>
<td>credit rating</td>
<td>F</td>
</tr>
<tr>
<td>credit rating</td>
<td>E</td>
</tr>
</tbody>
</table>

1. {a, h, t, f, No}
2. {a, h, t, e, No}
3. {b, h, t, f, Yes}
4. {c, m, t, f, Yes}
5. {c, l, s, f, Yes}
6. {c, l, s, e, No}
7. {b, l, s, e, Yes}
8. {a, m, t, f, No}
9. {a, l, s, f, Yes}
10. {c, m, s, f, Yes}
11. {a, m, s, e, Yes}
12. {b, m, t, e, Yes}
13. {b, h, s, f, Yes}
14. {c, m, t, e, No}

C1 and F1

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>a, Class=No</td>
<td>3</td>
</tr>
<tr>
<td>b, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>b, Class=No</td>
<td>0</td>
</tr>
<tr>
<td>c, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>c, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>h, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>h, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>m, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>m, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>l, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>l, Class=No</td>
<td>1</td>
</tr>
<tr>
<td>s, Class=Yes</td>
<td>6</td>
</tr>
<tr>
<td>s, Class=No</td>
<td>1</td>
</tr>
<tr>
<td>t, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>t, Class=No</td>
<td>4</td>
</tr>
<tr>
<td>f, Class=Yes</td>
<td>6</td>
</tr>
<tr>
<td>f, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>e, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>e, Class=No</td>
<td>3</td>
</tr>
<tr>
<td>h, s, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>h, t, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>h, t, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>m, s, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>m, t, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>m, t, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>l, s, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>l, t, Class=Yes</td>
<td>0</td>
</tr>
</tbody>
</table>

C2 and F2

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, s, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>a, t, Class=Yes</td>
<td>0</td>
</tr>
<tr>
<td>a, t, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>b, s, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>b, t, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>b, t, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>c, s, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>e, s, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>e, t, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>e, t, Class=No</td>
<td>4</td>
</tr>
<tr>
<td>a, f, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>a, f, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>a, e, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>a, e, Class=No</td>
<td>4</td>
</tr>
<tr>
<td>b, e, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>b, e, Class=No</td>
<td>0</td>
</tr>
<tr>
<td>c, f, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>c, e, Class=Yes</td>
<td>0</td>
</tr>
<tr>
<td>s, f, Class=Yes</td>
<td>3</td>
</tr>
<tr>
<td>s, e, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>t, f, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>t, f, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>t, e, Class=Yes</td>
<td>4</td>
</tr>
<tr>
<td>t, e, Class=No</td>
<td>0</td>
</tr>
</tbody>
</table>

If we map the attribute values as follows the training set becomes the following transactions:
C3 and F3

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>a, h, t, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>a, t, f, Class=No</td>
<td>2</td>
</tr>
<tr>
<td>b, s, e, Class=Yes</td>
<td>1</td>
</tr>
<tr>
<td>e, m, s, Class=Yes</td>
<td>0</td>
</tr>
<tr>
<td>c, m, f, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>c, s, f, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>m, s, Class=Yes</td>
<td>1</td>
</tr>
<tr>
<td>m, f, Class=No</td>
<td>1</td>
</tr>
<tr>
<td>l, s, f, Class=Yes</td>
<td>2</td>
</tr>
<tr>
<td>l, e, Class=Yes</td>
<td>1</td>
</tr>
</tbody>
</table>

No C4 candidates

Classification rules are:

a, h, t  Class=No : age_{<=30} AND income_{high} AND student_{no}  Class=No (14.3%, 100%)
a, t, f  Class=No : age_{<=30} AND student_{no} AND credit_rating_{fair}  Class=No (14.3%,100%)
c, m, f  Class=Yes : age_{>40} AND income_{medium} AND credit_rating_{fair}  Class=Yes (14.3%,100%)
l, s, f  Class=Yes : income_{low} AND student_{yes} AND credit_rating_{fair}  Class=Yes (14.3%,100%)
h, f  Class=Yes : income_{high} AND credit_rating_{fair}  Class=Yes (14.3%,66.6%)
m, f  Class=Yes : income_{medium} AND credit_rating_{fair}  Class=Yes (14.3%,66.6%) X
m, e  Class=Yes : income_{medium} AND credit_rating_{excellent}  Class=Yes (14.3%,66.6%)
l, f  Class=Yes : income_{low} AND credit_rating_{fair}  Class=Yes (14.3%,100%)
s, f  Class=Yes : student_{yes} AND credit_rating_{fair}  Class=Yes (21.4%,75%) X
s, e  Class=Yes : student_{yes} AND credit_rating_{excellent}  Class=Yes (14.3%,66.6%) X
h, t  Class=No : income_{low} AND student_{no}  Class=No (14.3%,66.6%)
m, s  Class=Yes : income_{high} AND student_{no}  Class=No (14.3%,66.6%) X
m, t  Class=Yes : income_{medium} AND student_{no}  Class=Yes (14.3%,100%) X
m, t  Class=No : income_{medium} AND student_{no}  Class=No (14.3%,50%)
l, s  Class=Yes : income_{low} AND student_{yes}  Class=Yes (21.4%,75%) X
a, f  Class=No : age_{<=30} AND credit_rating_{fair}  Class=No (14.3%,66.6%) X
b, e  Class=Yes : age_{31..40} AND credit_rating_{excellent}  Class=Yes (14.3%,100%)
c, f  Class=Yes : age_{<=40} AND credit_rating_{fair}  Class=Yes (21.4%,100%)
c, e  Class=No : age_{<=40} AND credit_rating_{excellent}  Class=No (14.3%,100%)
a, s  Class=Yes : age_{<=30} AND student_{yes}  Class=Yes (14.3%,100%) X
a, t  Class=No : age_{<=30} AND student_{no}  Class=No (14.3%,66.6%)
b, s  Class=Yes : age_{31..40} AND student_{yes}  Class=Yes (14.3%,100%)
b, t  Class=Yes : age_{<=40} AND student_{no}  Class=Yes (14.3%,100%)
c, s  Class=Yes : age_{<=40} AND student_{yes}  Class=Yes (14.3%,66.6%)
New example: age<=30, income=medium, student=yes, credit-rating=fair
There are 9 rules that apply (with red X above). Here they are ranked by confidence:

age<=30 AND student=yes \rightarrow Class=Yes (14.3\%,100\%)
income=medium AND student=yes \rightarrow Class=Yes (14.3\%,100\%)
student=yes \rightarrow Class=Yes (42.8\%,85.7\%)
student=yes AND credit_rating=fair \rightarrow Class=Yes (21.4\%,75\%)
credit_rating=fair \rightarrow Class=Yes (42.8\%,75\%)
income=medium \rightarrow Class=Yes (28.6\%,66.6\%)
age<=30 AND credit_rating=fair \rightarrow Class=No (14.3\%,66.6\%)
income=medium AND credit_rating=fair \rightarrow Class=Yes (14.3\%,66.6\%)
age<=30 \rightarrow Class=No (21.4\%,60\%)

The highest confident rules predicts Class=Yes.
We would in that case predict Buys_computer = yes

In a vote case:
There are 7 rules predicting Class=Yes with combined confidence = 81.27%
There are 2 rules predicting Class=No with combined confidence = 63.3%
We would in that case predict Buys_computer = yes