1. Consider the following decision tree:

(a) Draw the decision boundaries defined by this tree. Each leaf of the tree is labeled with a letter. Write this letter in the corresponding region of input space.

(b) Give another decision tree that is syntactically different but defines the same decision boundaries. This demonstrates that the space of decision trees is syntactically redundant. Is this redundancy a statistical problem (i.e., does it affect the accuracy of the learned trees)? Is it a computational problem (i.e., does it increase or decrease the computational complexity of finding an accurate tree)?

2. In the basic decision tree algorithm, we choose the feature/value pair with the maximum mutual information as the test to use at each internal node of the decision tree. Suppose we modified the algorithm to choose at random from among those feature/value combinations that had non-zero mutual information, but that we kept all other parts of the algorithm unchanged.

(a) Prove that if a splitting feature/value combination has non-zero mutual information at an internal node, then at least one training example must be sent to each of the child nodes.

(b) What is the maximum number of leaf nodes that such a decision tree could contain if it were trained on \(m\) training examples?

(c) What is the maximum number of leaf nodes that a decision tree could contain if it were trained on \(m\) training examples using the original maximum mutual information version of the algorithm? Is it bigger, smaller, or the same as your answer to (b)?

(d) How do you think this change would affect the accuracy of the decision trees produced on average? Why?
3. Consider the following training set:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

a. Learn a Naive Bayes classifier by estimating all necessary probabilities. Make prediction for \((A=1, B=0, C=0)\).

b. Suppose we know that \(A\), \(B\) and \(C\) are independent random variables, can we say that the Naive Bayes assumption is valid? (Note that the particular data set is irrelevant for this question). If your answer is yes, please explain why; if you answer is no please give an counter example.

d. Learn a decision tree from the training set shown above using the Mutual Information criterion.

5. Let's consider an alternative neural network diagram different from the lectures. In this diagram, a neural network uses a softmax activation function for its output layer. Its outputs can be interpreted as posterior probabilities \(P(y|x)\) for a categorical target variable \(y\). Consider the following neural network with three output units. The softmax activation function is defined as: 
\[
\hat{y}_i = \frac{\exp(x_i)}{\sum_{j=1}^{3} \exp(x_j)} \quad \text{(as opposed to what we saw in class \(\hat{y}_i = \frac{1}{1+\exp(-x_i)}\))}
\]
where \(x_i\) is the net input input the activation function of the output node \(i\). Note that \(\hat{y}_1 + \hat{y}_2 + \hat{y}_3 = 1\), making it a valid posterior probability. In this problem, we will compute the derivatives needed for the backpropagation algorithm for this kind of network.

```
    y1-hat       y2-hat       y3-hat
     |               |               |
  W9  |               |               | W11
     |               |               |
  a6  |               |               | a8
     |               |               |
  W6  |               |               | W8
     |               |               |
  1   |               |               | 1
     |               |               |
1 x1 x2 x3 x4
```

a. Write down the log likelihood objective function \(J(w)\) for this network, where \(w\) is the concatenation of \(W6, W7, W8, W9, W10,\) and \(W11\). You may assume that each training example has the form \((x, y)\), where \(x = (1, x_1, x_2, x_3, x_4)\) and \(y = (y_1, y_2, y_3)\). There are only three possible \(y\) values: \(y = (1, 0, 0), y = (0, 1, 0), \) and \(y = (0, 0, 1)\).
b. Compute the partial derivative 
\[ \frac{\partial J(w)}{\partial w_{9,6}} \]

c. Compute the partial derivative 
\[ \frac{\partial J(w)}{\partial w_{6,3}} \]

d. Generalize your answers to (b) and (c) and write the pseudo-code for the backpropagation algorithm using them.

6. [6] Cubic Kernels. In class, we showed that the quadratic kernel \( K(x_i, x_j) = (x_i \cdot x_j + 1)^2 \) was equivalent to mapping each \( x \) into a higher dimensional space where
\[
\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)
\]
for the case where \( x = (x_1, x_2) \). Now consider the cubic kernel \( K(x_i, x_j) = (x_i \cdot x_j + 1)^3 \). What is the corresponding \( \Phi \) function (again, for the special case where \( x = (x_1, x_2) \))? 

**Implementation assignment**

I. Implement the Naive Bayes classifier for categorical features with and without Laplace smoothing. Test your implementation on the provided data sets (train on the training data and test with the testing data sets).

**Data set information:** This data set is extracted from the UCI zoo data set. Note that there are 16 features (the first 16 columns) and the class labels are in the last column. There are 7 classes (numerically specified as class 1 to 7). All features are binary except for feature 13, which is a categorical variable with possible values 0, 2, 4, 5, 6, 8.

**Need to report:**
1. Report the parameters that are learned by your classifiers (with and without Laplace smoothing respectively).
2. Report the test set accuracy for both cases.

II. Implement a fixed depth decision tree algorithm. In particular, the input to your algorithm will include the training data set and the maximum depth of the tree. For example, if the depth is set to one, you will learn a decision tree with one test node, which is also called a decision stump. Test your implementation, with depth=1, and 2 respectively, on the same data set as described above (train on the training data and test with the testing data set).

**Need to report:**
1. Report the learned decision tree (depth 1 and depth 2)
2. Report the test set accuracy for both trees.