The exam is closed book. You are allowed a one-page cheat sheet. Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, use an additional sheet (available from the instructor) and staple it to your exam.

- NAME ____________________________________________
- UTD-ID if known ____________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Trees</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Linear Classifiers and Regularization</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Neural Networks and Perceptrons</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>KD-trees and Short questions</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>65</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: Decision Trees (15 points)
Consider the training dataset given below. $X_1$, $X_2$, $X_3$ and $X_4$ are the attributes/features and $Y$ is the class variable.

<table>
<thead>
<tr>
<th>Y</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>+1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) (8 points) Greedily learn a decision tree using the ID3 (namely attributes selected using the information gain criteria) algorithm. Do the calculations on the rough sheets provided by the instructor and draw the tree below.
(b) (5 points) Draw a decision tree having only 4 leaf nodes, 3 internal nodes and depth bounded by 2, that has 100% accuracy on the given dataset.

(c) (2 points) Which decision tree will you prefer: (a) the ID3 tree you drew on the previous page or (b) the tree you drew on this page having 4 leaf nodes and 3 internal nodes. Also, explain in one or two sentences why you will prefer the tree (you said you will prefer)? No credit without a correct explanation.
Question 2: Linear Classifiers and Regularization (10 points)

(a) (4 points) Recall that a linear threshold function or a linear classifier is given by:

If \((w_0 + \sum_i w_i x_i) > 0\) then class is positive, otherwise it is negative. Assume that 1 is true and 0 is false.

Consider a function over \(n\) Binary features, defined as follows. If at least \(k\) variables are false, where \(k \leq n\) is a constant, then the class is positive, otherwise the class is negative. Can you represent this function using a linear threshold function. If your answer is YES, then give a precise numerical setting of the weights. Otherwise, clearly explain, why this function cannot be represented using a linear threshold function.
In this problem, we will refer to the binary classification task depicted in the figure given below. Consider the following logistic regression (LR) model:

\[
P(y = 1|x_1, x_2, w_1, w_2) = \frac{1}{1 + \exp(w_1 x_1 + w_2 x_2)}
\]

Notice that the model is assuming that the bias term \( w_0 \) equals 0, namely the induced classifiers will pass through the origin.

In the figure, the dotted lines are the axes. \( x_1 \) is the \( X \)-axis and \( x_2 \) is the \( Y \)-axis.

Let \( L1 \) be the solution (line) output by a gradient ascent algorithm using the maximum likelihood estimation (MLE) criteria. In the following, you will answer how regularization will impact the solution.

(b) (6 points) Consider a regularization approach where we try to maximize:

\[
\sum_{i=1}^{n} \log \{ P(y_i|x_{i,1}, x_{i,2}, w_1, w_2) \} - \frac{C}{2} (w_1)^2
\]

for large \( C \). Note that only \( w_1 \) is penalized. We’d like to know which of the lines in the figure above could arise as a result of such regularization. For each potential line \( L2, L3 \) or \( L4 \) determine whether it can result from regularizing \( w_1 \). If not, explain very briefly why not.

- L2 (answer Yes/NO and briefly explain why):
• L3 (answer Yes/NO and briefly explain why):

• L4 (answer Yes/NO and briefly explain why):
Question 3: Neural Networks and Perceptrons (20 points)

(a) (10 points) Draw a neural network that represents the following function. You can only use two types of units: linear units and sign units. Recall that the linear unit takes as input weights and attribute values and outputs \( w_0 + \sum_i w_i x_i \), while the sign unit outputs +1 if \( (w_0 + \sum_i w_i x_i) > 0 \) and −1 otherwise.

\( y_1 \) and \( y_2 \) are outputs and \( x_1 \) and \( x_2 \) are inputs. Therefore, your neural network will have two output nodes.

\[
\begin{array}{|c|c|c|c|}
\hline
x_1 & x_2 & y_1 & y_2 \\
\hline
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 \\
\hline
\end{array}
\]

Note that to get full credit, you have to write down the precise numeric weights (e.g., −1, −0.5, +1, etc.) as well as the precise units used at each hidden and output node.
(b) (10 points) Derive a gradient descent training algorithm for the following “special” unit and evaluation function. The “special” unit takes as input a vector \((x_1, \ldots, x_n)\) of feature values and outputs \(o\), where \(o\) is given by the following equation:

\[
o = w_0 + \sum_{j=1}^{n} w_j (x_j + x_j^{1.5}) (1)
\]

Here, \(w_0, w_1, \ldots, w_n\) are the parameters which you have to learn from the training dataset \(D\) having \(m\) examples. Use the following evaluation (error) function:

\[
E = \frac{1}{3} \sum_{i=1}^{m} (y_i - o_i)^3
\]

where \(y_i\) is the value of the \(i\)-th example that your algorithm will predict and \(o_i\) is given in equation (1). Use the batch gradient descent approach. Use the following notation: \(x_{i,d}\) denotes the value of the \(i\)-th attribute (feature) in the \(d\)-th example in the training set \(D\).
Question 4: Support Vector Machines (10 points)

Consider the dataset given below \((x_1, x_2)\) are the attributes and \(y\) is the class variable:

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>−2</td>
<td>1</td>
<td>−1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>−1</td>
</tr>
</tbody>
</table>

(a) (10 points) Find the linear SVM classifier for the dataset given above. Do your optimization either using the primal problem or the dual problem. Provide a precise setting of the weights \(w\) and the bias term \(b\). What is the size of the margin? (Hint: The primal problem seems to be easier than the dual).
Consider the set of 2-D points along with the co-ordinates given above:

(a) (3 points) Build a balanced kd-tree for the points given above. Draw the tree below and the separating planes in the figure.

(b) (3 points) Make another copy of the tree and highlight the edges traversed when looking for the closest point to \((5, 8)\).
(a) (2 points) Yes/No. For linearly separable data, can a small slack penalty hurt the training accuracy when using a linear SVM (no kernel)? If so, explain how. If not, why not?

(b) (2 points) Provide a reasonable approach for determining the value of $K$ in the $K$-nearest neighbors algorithm.