Instructions
- Answer 5 out of 6 questions from the first section (i.e., True / False questions).
- Answer 2 out of 3 questions from the second section.
- Total time 3 hours.
- All written or printed material is allowed.
- Explain your steps.
- Good luck!

1. True/False questions [40 pt]:
   (a) The $k$-means algorithm is a Lazy-Learning algorithm. [8 pt]
   (b) The Perceptron algorithm is a special case of the SVM algorithm where the regularization factor is 0 ($\lambda = 0$), and the learning rate is 1 ($\eta = 1$). [8 pt]
   (c) PCA can be used to increase the data's dimensionality. [8 pt]
   (d) In DNN, we use a non-linear activation function in order to allow a given neural network to learn complex non-linear separable tasks [8 pt]
   (e) If we wish to predict whether a patient has the KAZABOBO disease or not given her lab test results we can use the Perceptron algorithm, since it's a binary classification learning task. [8 pt]
   (f) The main idea of the kernel function is to map the data that are linearly inseparable in the observation space into a higher dimensional space where they are linearly separable. [8 pt]

2. Logistic Regression [30 pt]
In this question, we will introduce a different approach to the classification problem, called Logistic Regression. Assume that $y \in \{0, 1\}$. We define the conditional probability as follows:

$$P_w(x|y = 1) = \sigma(w \cdot x) = \frac{1}{1 + e^{-w \cdot x}}, \quad (1)$$

Notice that the returned value is indeed between 0 and 1.
We define the likelihood as follows:

\[ L(w) = \sum_{i=1}^{m} P_w(x|y = 1)^{y_i} (1 - P_w(x|y = 1))^{1-y_i} \] (2)

We wish to find the weight vector \( w \) that would maximize the likelihood or equivalently would minimize the negative log likelihood:

\[ l(w) = -\sum_{i=1}^{m} y_i \log P_w(x|y = 1) + (1 - y_i) \log (1 - P_w(x|y = 1)) \] (3)

\[ w^* = \arg\min_w l(w) \] (4)

To minimize the objective, we will use a derivation similar to the one we used for linear regression; gradient descent. Written in a vectorial notation, the update rule is given by:

\[ w_{t+1} = w_t - \eta \nabla_w l(w) \] (5)

(a) Find the derivative of the sigmoid function \( \sigma(z) \). [5 pt]
(b) Find the derivative \( \nabla_w l(w) \) when working with a single training example \((x, y)\). The derivative you found for \( \sigma(z) \) could be useful here. [5 pt]
(c) Finally, find the full gradient descent update rule. [10 pt]
(d) In both Logistic Regression and Linear Regression, given an input \( X \), the goal is to predict the response \( Y \). The difference is that Logistic Regression is typically used for classification whereas Linear Regression is used for regression.

Propose a simple modification to Linear Regression, that makes it amenable to classification (instead of regression) tasks. Comment on whether such a proposal is superior than Logistic Regression or not and briefly explain why. [10 pt]

3. Large Margin Perceptron Algorithm [30 pt]
A modified version of the Perceptron algorithm is given in Figure 1.

```
INIT: training set \( S = \{(x_i, y_i)\}_{i=1}^{m} \)
INITIALIZE: \( w = 0 \)
LOOP:
  • choose example \((x_i, y_i)\) uniformly at random from \( S \)
  • let \( \ell(w, x_i, y_i) = 1 - y_i w \cdot x_i \)
  • if \( \ell(w, x_i, y_i) > 0 \):
    • update: \( w = w + y_i x_i \)
```

Figure 1: Large margin Perceptron
(a) Why this algorithm is called Large Margin Perceptron? [5 pt]

(b) Assume that $\|x_i\| \leq R$ for all $i$ and that there exists a vector $u$, $\|u\| = 1$ such that $y_i(u \cdot x_i) \geq \gamma$ for all $i$. Derive an upper bound for the number of mistakes made by this large margin Perceptron algorithm. [10 pt]

(c) Is this upper bound on the number of mistakes smaller or greater than the upper bound of the standard Perceptron algorithm? [10 pt]

(d) Based on your answer to (c) - what is the role of the margin here? would you prefer to use this algorithm over the standard Perceptron algorithm when your data noisy? [5 pt]

4. **Decision Tree** [30 pt]
Our goal is to construct a decision tree classifier for predicting flight delays. We have collected data for a few months and a summary of the data is provided in the following table.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value = yes</th>
<th>Value = no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>Delayed - 30, not Delayed - 10</td>
<td>Delayed - 10, not Delayed - 30</td>
</tr>
<tr>
<td>Wind</td>
<td>Delayed - 25, not Delayed - 15</td>
<td>Delayed - 15, not Delayed - 25</td>
</tr>
<tr>
<td>Summer</td>
<td>Delayed - 5, not Delayed - 35</td>
<td>Delayed - 35, not Delayed - 5</td>
</tr>
<tr>
<td>Winter</td>
<td>Delayed - 20, not Delayed - 10</td>
<td>Delayed - 20, not Delayed - 30</td>
</tr>
<tr>
<td>Day</td>
<td>Delayed - 20, not Delayed - 20</td>
<td>Delayed - 20, not Delayed - 20</td>
</tr>
<tr>
<td>Night</td>
<td>Delayed - 15, not Delayed - 10</td>
<td>Delayed - 25, not Delayed - 30</td>
</tr>
</tbody>
</table>

(a) Describe the method for determining the root of the decision tree. [6 pt]

(b) Based on the table, can you determine the root of the decision tree without computing exact values? if so, name the feature. [6 pt]

(c) Based on the table, can you determine which feature should be on the second level (the level just beneath the root) without computing exact values? if so, name the feature. [6 pt]

(d) Real datasets may not be perfect, e.g. some may contain systematic errors. In each of these situations, if there are systematic errors, describe how you detect them and what we should do. [6 pt]

1. An attribute $I$ has only one single value.
2. Attributes $T_1$ and $T_2$ are duplicates. That means all examples have the same values for these two attributes.

(e) Alice and Bob argued about the nature of the decision tree. Alice said that after a training process over a binary dataset the output decision tree is clearly a linear classifier. Bob disagreed. Which of the is correct? explain. [6 pt]