Dry Part

Question 1

We learned that given a space R^d the vector $w \in R^d$ defines a homogeneous halfspace of the space using the classification function $f(x) = sign(\langle x, w \rangle)$

We would like to find a function $g : R^d \rightarrow R^{d+1}$ which will define a non-homogeneous halfspace of R^d.

Therefore, for all $x \in R^d$ such that $x = (x_1, x_2, \dots, x_d)$ we will define $g : R^d \rightarrow R^{d+1}$ by $g(x) = (1, x_1, x_2, \dots, x_d)$.

For all super-plane which induces a non-homogeneous halfspace of R^d, that is defined by $z \in R^{d+1}$ we can take the value z_1, which is actually can be seen as moving the first coordinate by z_1 from the origin.

Therfore, there exists a vector $w \in R^d$ such that $w = (z_2, z_3, \dots, z_{d+1})$ which defines a super-plane that induces a non-homogeneous halfspace of R^d, and is parallel to the plane which is defined by z.

Thus, $f(x) = sign(\langle g(x), z \rangle)$ defines the non-homogeneous halfspace.

Question 2

For dataset A:

1. Perceptron - cannot fit perfectly as the dataset is not linearly seperable. You might be able to separate the data in a higher dimension by adding features.
2. KNN with k=1 - can fit perfectly as each sample in training set will be classified by itself
3. KNN with k=3 - might not fit perfectly as there are samples close enough to two other samples from the other color.
4. Decision Tree - can fit perfectly as you can separate the data into 4 quadrons which can be labeled using only 4 leafs.

For Dataset B:

1. Perceptron - can fit perfectly, as the dataset is clearly linearly seperable.
2. KNN with k=1 - can fit perfectly as each sample in training set will be classified by itself
3. KNN with k=3 - cannot fit perfectly as there are samples which are close to two other samples from the other color, and as such they will be classified wrongfuly.
4. Decision Tree - cannot fit perfectly, because you'll need more than 4 leafs to separate the data on the "diagonal" between classes

For Dataset C:

1. Perceptron - cannot fit perfectly, as the dataset is not linearly seperable. You might be able to separate the data in a higher dimension by adding features.
2. KNN with k=1 - can fit perfectly as each sample in training set will be classified by itself

3. KNN with k=3 - can fit perfectly because all samples are only close to two other samples by the same color, and as such they will be classified correctly.
4. Decision Tree - cannot fit perfectly on the given features because you'll need more than 4 leafs the seperate the "circle" which is not a linear horizontal or vertical line. If we'll add a feature $x^2 + y^2$ we can use a decision tree (and perceptron for that matter).

Question 3

Part 1:

We will derive $\frac{\partial}{\partial w_j} l(x_i, y_i, w)$, using the known sigmoid derviative $\sigma(z)' = \sigma(z)(1 - \sigma(z))$

$$\frac{\partial}{\partial w_j} l(x_i, y_i, w) = \frac{\partial}{\partial \sigma} l(x_i, y_i, w) \cdot \frac{\partial}{\partial w_j} \sigma(y_i \langle x_i, w \rangle)$$

$$= \frac{1}{\sigma(y_i \langle x_i, w \rangle)} \cdot \frac{\partial}{\partial w_j} \sigma(y_i \langle x_i, w \rangle)$$

$$= \frac{1}{\sigma(y_i \langle x_i, w \rangle)} \cdot \sigma(y_i \langle x_i, w \rangle) \cdot (1 - \sigma(y_i \langle x_i, w \rangle)) \cdot \frac{\partial(yi \cdot \langle x_i w \rangle)}{\partial w_j}$$

$$= (1 - \sigma(y_i \langle x_i, w \rangle)) \cdot y_i \cdot x_{i,j}$$

Part 2:

We will derive w.r.t w using the above derviative:

$$\frac{\partial}{\partial w} l(x_i, y_i, w) = \begin{bmatrix} \frac{\partial}{\partial w_1} l(x_i, y_i, w) & \dots & \frac{\partial}{\partial w_n} l(x_i, y_i, w) \end{bmatrix}$$

$$= [\, (1 - \sigma(y_i \langle x_i, w \rangle)) \cdot y_i \cdot x_{i,1} \quad \dots \quad (1 - \sigma(y_i \langle x_i, w \rangle)) \cdot y_i \cdot x_{i,n} \,]$$

$$= (1 - \sigma(y_i \langle x_i, w \rangle)) \cdot y_i \cdot x_i$$