Manuel Ortiz Cachá (k20114995)

7CCSMPNN Pattern Recognition, Neural Networks and Deep Learning

Coursework Week 9: Ensemble Methods

Due: Thursday 22nd April 2021 (23:59 UK time)

This coursework is assessed. A type-written report needs to be submitted online through KEATS by the deadline specified on the module's KEATS webpage. This coursework considers your "own created" dataset to investigate the classification performance using the techniques of Bagging and Boosting. Some simple "weak" classifiers will be designed and combined to achieve an improved classification performance for a two-class classification problem.

INTRODUCTION: Ensemble Methods

The key idea behind the Ensemble Methods is to generate a collection of "weak" classifiers from a given dataset that increases the overall accuracy or performance. This can be necessary for datasets that are very large or have a big feature space. At the end, the purpose is to combine the "weak" classifiers somehow (with a serial or parallel structure) to achieve a final one, by averaging or voting.

The "weak" classifiers are simple classifiers that perform slightly better than a half-space or random classifier ($accuracy > 50\%$). These classifiers are generated from sections of the given dataset, by resampling. Basically, we break the dataset in a number of simple datasets, and then we apply a "weak" classifier to each one to obtain an overall strong classifier.

This kind of methods imply numerous advantages:

- It reduces variance, improving the precision and the overall classifier's performance. Results are less dependent on peculiarities of a single training set.
- It reduces bias, obtaining a more accurate classifier.
- It improves the estimate if the learning algorithm is unstable. By unstable we mean that small changes in the training set imply large changes in the output classifier. This could be said about Neural Networks or Decision Trees.
- In short, the goal and motivation behind this technique is to improve the accuracy by using multiple classifiers. These methods attempt to avoid overfitting or underfitting the data, obtaining a clean accurate classifier.

In the last coursework, the one about multi-class SVMs, we had a taste of ensemble methods. In my personal case, I solved a multi-class problem with a Binary Decision Tree SVM approach, which in essence, was an ensemble of two binary SVMs.

There are different techniques for creating an ensemble model, in this coursework we will dig deeper into the Bagging and Boosting methods, dedicating one question to each. Among others, we also have the AdaBoost (Boosting variation) algorithm or Random Forests.

Q1. Create a non-linearly separable dataset consisting of at least 20 two-dimensional dataset. Each data is characterised by two points $x_1 \in [-10, 10]$ and $x_2 \in [-10, 10]$ and associated with a class $y \in \{-1, +1\}$. List the data in a table in a format as shown in Table 1 where the first column is for the data points of class "-1" and the second column is for the data points of class "$+1$". (20 Marks)

In order to create a non-linearly separable dataset, we must create two datasets that cannot be separated by a straight line. Hence, if we manage to create an inseparable dataset, we have managed to create a non-linearly separable dataset.

We have little restrictions: $x_1 \in [-10, 10]$; $x_2 \in [-10, 10]$; the dataset has to contain at least 20 instances, 10 for each class; and these two classes must be non-linearly separable. With these requirements we have built the following dataset:

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-7, 0)	(0, 4)
2	(-5, 5)	(1, 1)
3	(-4, -7)	(1, 3)
4	(-3, 9)	(2, 3)
5	(-2, 6)	(2, 5)
6	(0, -6)	(3, 2)
7	(4, 9)	(4, 1)
8	(2, 8)	(4, 3)
9	(-7, -7)	(4, 4)
10	(-9, -1)	(5, 2)

Table 1: Dataset of two classes

In the following plot we can see these data samples plotted. As it can be inferred graphically, there is no straight line that can divide the two data classes. Therefore, it is said that the dataset is non-linearly separable.

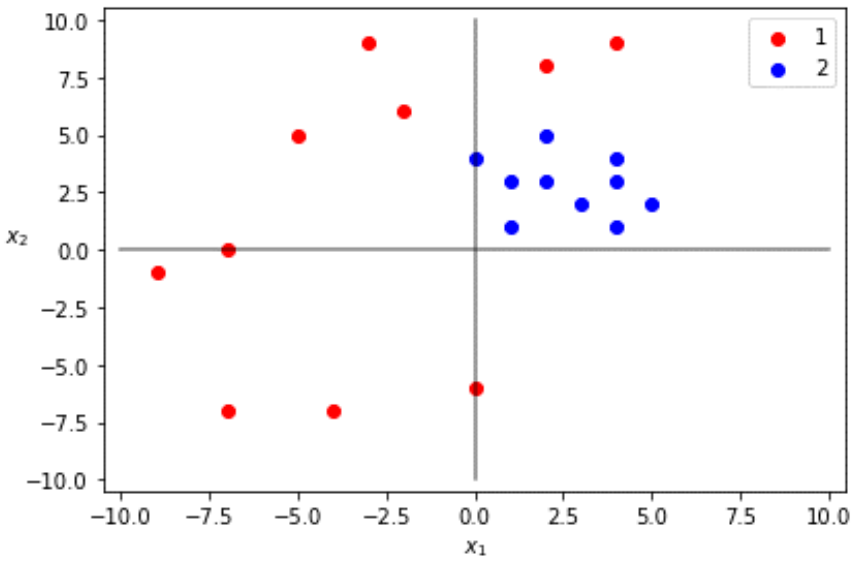

Figure 1: Dataset of two classes

Q2. Plot the dataset (x axis is x_1 and y axis is x_2) and show that the dataset is non-linearly separable. Represent class "-1" and class "$+1$" using "$\times$" and "$\circ$", respectively. Explain why your dataset is non-linearly separable. *Hint: the Matlab built-in function plot can be used.* (20 Marks)

We have already plotted the data instances in the previous question. We will plot them again with the naming indicated in the question description.

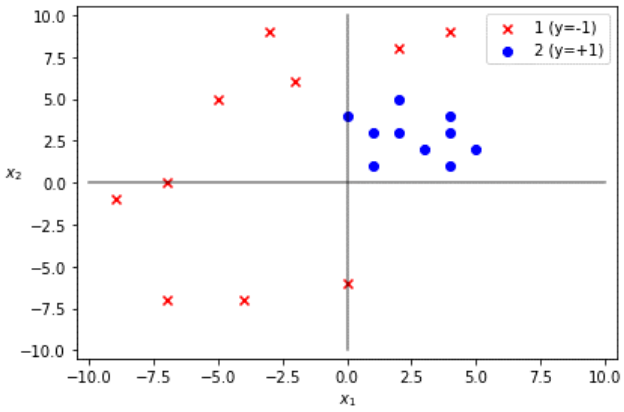

Figure 2: Dataset of two classes

Now we have to explain why this dataset is not linearly separable. From the definition, we have that in Euclidean geometry, linear separability is a property of two sets of points. This is most easily visualised in two dimensions (the Euclidean plane) by thinking of one set of points as being coloured blue and the other set of points as being coloured red. These two sets are linearly separable if there exists at least one line in the plane with all of the blue points on one side of the line and all the red points on the other side. This idea immediately generalizes to higher-dimensional Euclidean spaces if the line is replaced by a hyperplane. [1]

As we can see in *Figure 2*, there is no possible line that can divide the data points from class 1 from the data points from class 2. We could draw some lines (see *Figure 3*) that isolate all samples from class 2 (blue circles), but there will be also some instances from class 1 (red crosses) on that side of the line too. Therefore, it is not linearly separable as the two classes cannot be separated by a linear line.

Here are some examples that partially separate the data classes:

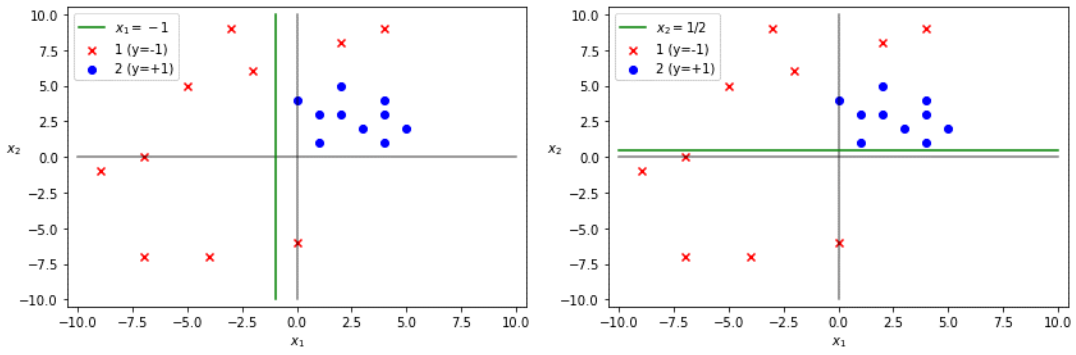

Figure 3: Proof of non-linearity

[1] https://en.wikipedia.org/wiki/Linear_separability

Q3. Design Bagging classifiers consisting of 3, 4 and 5 weak classifiers using the steps shown in Appendix 1. A linear classifier should be used as the weak classifier. Explain and show the design of the hyperplanes of weak classifiers. List the parameters of the design hyperplanes.

After designing the weak classifiers, apply the designed weak classifiers and bagging classifier to all the samples in Table 1. Present the classification results in a table as shown in Table 2. The columns "Weak classifier 1" to "Weak classifier n" list the output class $(\{-1, +1\})$ of the corresponding weak classifiers. The column "Overall classifier" list the output class $(\{-1, +1\})$ of the bagging classifier. The last row lists the classification accuracy in percentage for all classifiers, i.e., $\frac{\text{Number of correct classifications}}{\text{Total number of samples}} \times 100\%$. Explain how to determine the class (for each weak classifier and over all classifier) using one test sample. You will have 3 tables (for 3, 4 and 5 weak classifiers) for this question. Comment on the results (in terms of classification performance when different number of weak classifiers are used). (30 Marks)

We will start by designing the weak classifiers of the Bagging method. The name Bagging comes from "bootstrap aggregating". The procedure to carry out a Bagging ensemble model are presented in the Appendix 1 of the Coursework description:

- STEP 1: start with dataset $\mathcal{D}$. We have our dataset defined in Q1 and plotted in Q2.
- STEP 2: generate *M* dataset $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M$.
    - Each distribution is created by drawing $n' < n$ samples from $\mathcal{D}$ with *replacement.*
    - Some samples can appear more than once while others do not appear at all.
- STEP 3: learn weak classifier for each dataset.
    - Weak classifier $f_i(x)$ for dataset $\mathcal{D}_i, i = 1, 2, \ldots, M$.
- STEP 4: combine all weak classifiers using a majority voting scheme.
    - $f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right)$

Once we know the steps we must follow, we can begin creating the Bagging classifier. We have to create an ensemble model of 3, 4, and 5 weak classifiers. Therefore, we will first create 3 weak classifiers by the procedure explained before, then we will only create one weak classifier more for the Bagging method with 4 linear classifiers, and lastly, we will generate another weak classifier for the Bagging classifier with 5 weak classifiers.

As a side note, in the Appendix 1 it literally says to "combine all weak classifiers using a majority voting scheme" in the final step (STEP 4). After that, it shows the following equation:

$$f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right)$$

I just wanted to clarify that the equation actually corresponds to the averaging scheme. In particular case where there are 3 or 5 weak classifiers this will give us the same outcome as majority voting. In any case, I will be using the averaging scheme, the one showed in the previous equation.

BAGGING ENSEMBLE MODEL WITH 3 WEAK CLASSIFIERS

The first step is to generate M $(M = 3)$ datasets by drawing $n' < n$ samples from $\mathcal{D}$ with *replacement*. In our case, we will choose n' to be 8 for each sub-dataset; which, as specified, is less than $n = 20$. To create this dataset we will help ourselves with the *.sample()* built-in function from Python. We will randomly choose 8 instances for the three different sub-datasets by simply passing 8 as the unique parameter.

As it is explained in the Appendix 1, we have to select these instances with replacement. This means that the three sub-datasets ($\mathcal{D}_1, \mathcal{D}_2$ and $\mathcal{D}_3$) may contain repeated instances, and some other instances may not appear in any of these three sub-datasets.

When a sampling unit is drawn from a finite population and is returned to that population, after its characteristic(s) have been recorded, before the next unit is drawn, the sampling is said to be "*with replacement*". In the contrary case the sampling is "*without replacement*". [2]

To better picture the instances randomly selected, I will show them in a table and then plot them. This way it is easier to imagine them. I will directly display the 8 data instances of each sub-dataset, without showing the code, which is straightforward. This would correspond to the STEP 2 of the procedure shown before.

1. $\mathcal{D}_1$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-7, 0)	(4, 4)
2	(0, -6)	(5, 2)
3	(-5, 5)	(1, 3)
4	(2, 8)	(0, 4)

Table 2: instances for sub-dataset $\mathcal{D}_1$

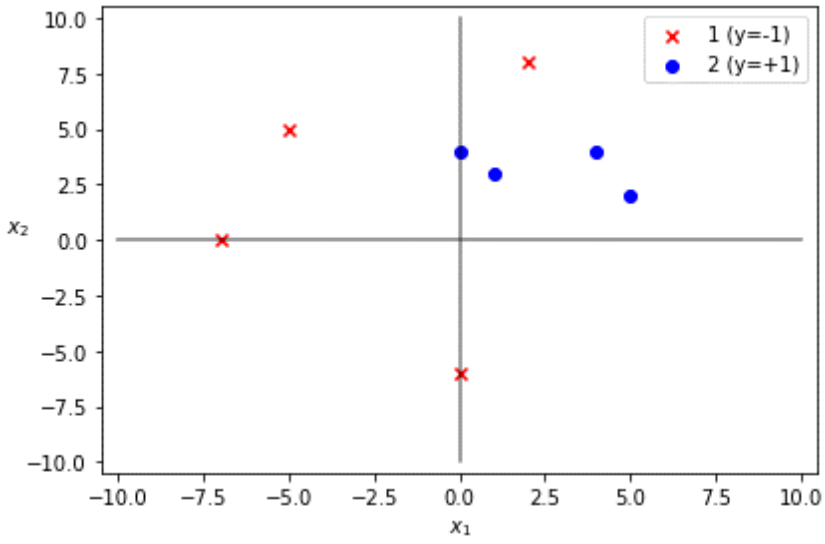

Figure 4: instances for sub-dataset $\mathcal{D}_1$

[2] https://stats.oecd.org/glossary/detail.asp?ID=3835

2. $\mathcal{D}_2$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-7, 0)	(4, 4)
2	(-3, 9)	(2, 3)
3	(2, 8)	(5, 2)
4	(4, 9)	(4, 1)

Table 3: instances for sub-dataset $\mathcal{D}_2$

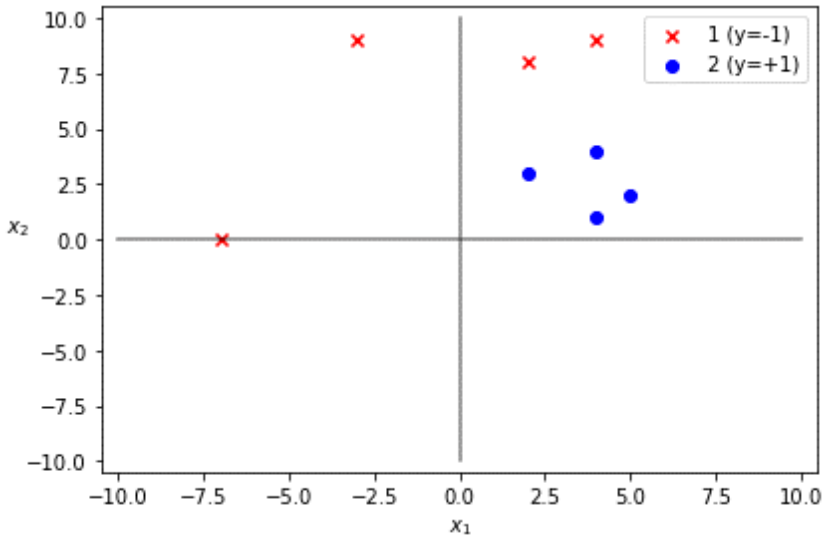

Figure 5: instances for sub-dataset $\mathcal{D}_2$

3. $\mathcal{D}_3$: as we can see, for this sub-dataset, the built-in function *.sample()* has randomly selected 5 instances from class 1 and 3 from class 2. There is no problem with this, the data samples need not be perfectly balanced.

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-3, 9)	(1, 3)
2	(0, -6)	(5, 2)
3	(-7, 0)	(4, 1)
4	(2, 8)	N/A
5	(-7, -7)	N/A

Table 4: instances for sub-dataset $\mathcal{D}_3$

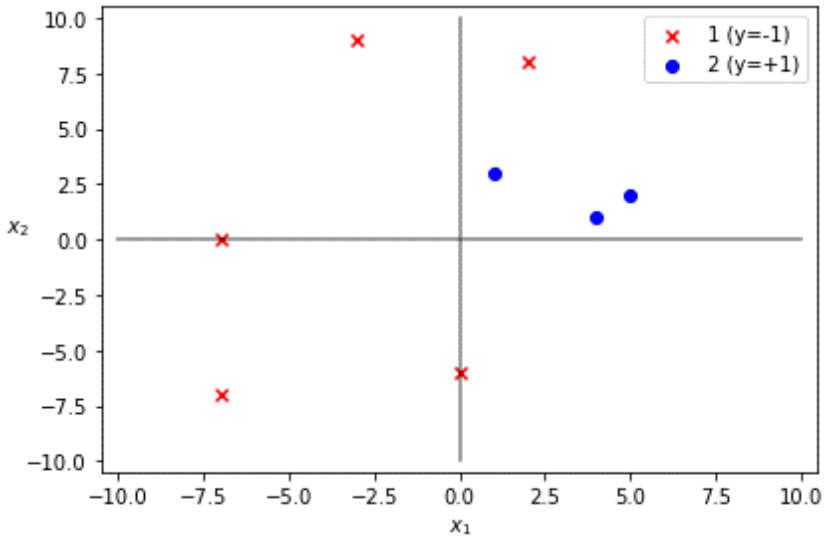

Figure 6: instances for sub-dataset $\mathcal{D}_3$

As we can see from the previous tables, there are a couple of instances that appear in every sub-dataset ((-7, 0), (2, 8) and (5, 2)). While, on the other hand, there are a lot of instances that do not appear at all in any sub-dataset, like (-4, -7), (3, 2) or (1, 1). This is normal with replacement.

The next stage we must accomplish is to learn the weak classifiers for each dataset (STEP 3). It is important to acknowledge that a weak classifier is a simple classifier, e.g., a half-space classifier, (has to be slightly better than choosing randomly/blindly), given by a different dataset (generated by resampling). This means that the weak classifiers that we will create must have an accuracy higher than 50% while classifying data instances. If not, it would not work and should be discarded.

We will generate each weak classifier by inspection, meaning that we will select a straight line that will define our weak classifier by observing the data instances. This procedure can also be called “by observation”.

The question description asks us to list the parameters of the design hyperplanes. The weak classifiers must be linear classifiers: straight lines in a 2-dimensional feature space. The parameters of each weak classifier will be the following:

$$f_i(x) \equiv x_2 = m \cdot x_1 + b$$

Where *m* and *b* are the parameters, *m* is the slope of the line, and b is the x_2-intercept. We will obtain these weak classifiers by applying the equation of a line passing through two points $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$:

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

Therefore, for each sub-dataset ($\mathcal{D}_1, \mathcal{D}_2$ and $\mathcal{D}_3$) we have a different weak classifier ($f_1(x), f_2(x)$ and $f_3(x)$).

Now, we are going to begin with $\mathcal{D}_1$ and its corresponding weak classifier, $f_1(x)$. The data representation is already seen on *Figure 4*, I will plot it again:

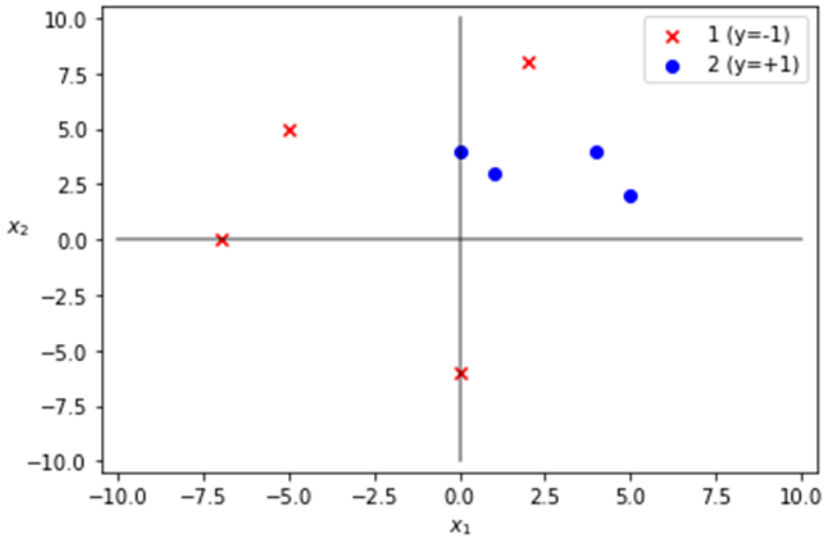

Figure 7: instances for sub-dataset $\mathcal{D}_1$

We could choose as our weak classifier a straight line with negative slope -2 that passes through the origin. For this, we could use the data points (-1, 2) and (1, -2). Substituting in the equation shown before, we would have the following:

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

$$x_2 - 2 = \frac{-2 - 2}{1 - (-1)} \times (x_1 - (-1)) \rightarrow x_2 = -2x_1$$

Therefore, we would have $f_1(x) \equiv x_2 = -2x_1$, where the slope is $m = -2$ as we said, and the intercept with the x_2-axis is $b = 0$. This weak classifier represented in a 2-dimensional plot would look something like this:

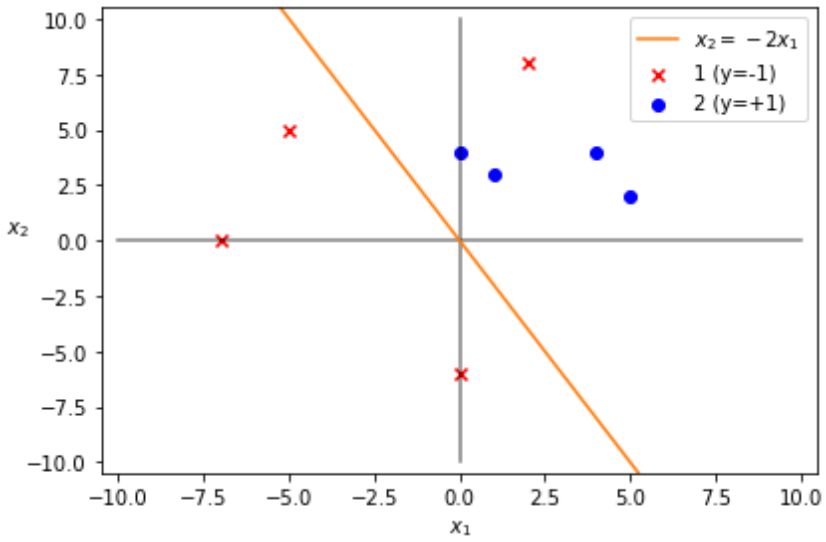

Figure 8: 2D representation of the weak classifier $f_1(\mathrm{x})$

We already have the straight line, now we need the weak classifier per se. This classifier will have the following form:

$$f_1(x) = \begin{cases} +1, & if\ x_2 + 2x_1 \geq 0 \\ -1, & otherwise \end{cases}$$

Now we will determine the classification accuracy to check if it is above 50%, if not, we should discard it.

INSTANCE	ACTUAL CLASS	OUTPUT OF $f_1(x)$	CORRECT CLASSIFICATION
(-7, 0)	-1	-1	YES
(0, -6)	-1	-1	YES
(-5, 5)	-1	-1	YES
(2, 8)	-1	+1	NO
(4, 4)	+1	+1	YES
(5, 2)	+1	+1	YES
(1, 3)	+1	+1	YES
(0, 4)	+1	+1	YES

Table 5: Classification error for the first dataset ($\mathcal{D}_1$) and the first weak classifier (f_1(x))

Looking at the previous table, *Table 5*, we can calculate the overall classification accuracy. This is the percentage of total correct predictions divided by the total number of instances. In our case, it is the following:

$$Accuracy\ for\ f_1(x) = \frac{7}{8} \times 100 = 87.5\% > 50\%\ (OK)$$

Hence, this weak classifier suits us well, and we can conserve it.

Next, we will do the same process with $\mathcal{D}_2$, choose by observation another straight line that separates data instances from class 1 and 2. Just like before, this weak classifier must have an accuracy above 50%. We will begin by observing the data instances from $\mathcal{D}_2$ again:

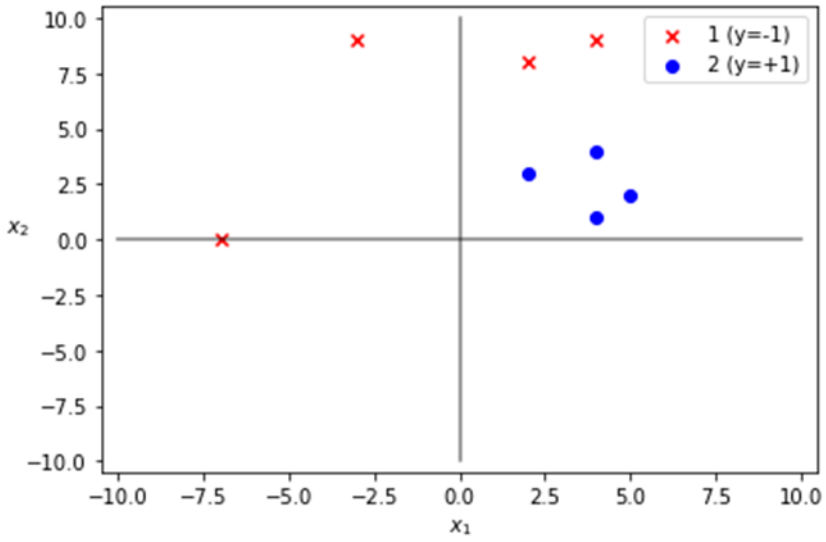

Figure 9: instances for sub-dataset $\mathcal{D}_2$

For this dataset we will choose a straight line parallel to the horizontal axis, x_1, that passes through $x_2 = 5$. Therefore, the straight line from $f_2(x)$ will have a slope $m = 0$ and the intercept of the the x_2-axis will be $b = 5$. We end up with the following line equation:

$$f_2(x) \equiv x_2 = m \cdot x_1 + b \rightarrow x_2 = 5$$

The following plot represents this weak classifier, $f_2(x)$:

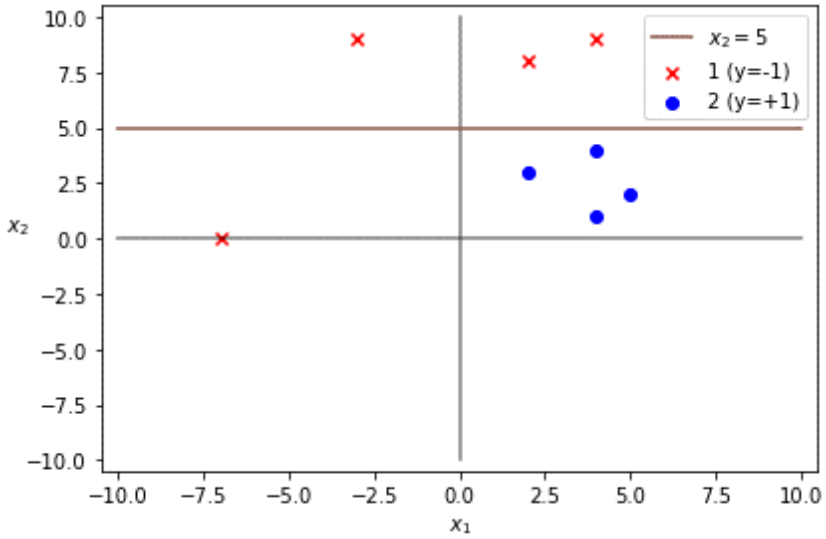

Figure 10: 2D representation of the weak classifier $f_2(x)$

This second weak classifier will have the following final form:

$$\boldsymbol{f_2(x) = \begin{cases} +1, & \textit{if } x_2 \leq 5 \\ -1, & \textit{otherwise} \end{cases}}$$

Now, just like before, we will see the accuracy of this classifier to verify that it suits us. We will create a table just like the one shown before:

INSTANCE	ACTUAL CLASS	OUTPUT OF $f_1(x)$	CORRECT CLASSIFICATION
(-7, 0)	-1	+1	NO
(-3, 9)	-1	-1	YES
(2, 8)	-1	-1	YES
(4, 9)	-1	-1	YES
(4, 4)	+1	+1	YES
(2, 3)	+1	+1	YES
(5, 2)	+1	+1	YES
(4, 1)	+1	+1	YES

Table 6: Classification error for the second dataset ($\mathcal{D}_2$) and the second weak classifier ($f_2(x)$)

Therefore, the overall accuracy for $f_2(x)$ is:

$$Accuracy\ for\ f_2(x) = \frac{7}{8} \times 100 = 87.5\% > 50\%\ (OK)$$

We can conserve it as its accuracy is above 50%.

Lastly, we have to design the last weak classifier, $f_3(x)$, which we will do so with the sub-dataset $\mathcal{D}_3$. This sub-dataset represented in a 2D plot would look like the one in *Figure 6*, which looks like the following:

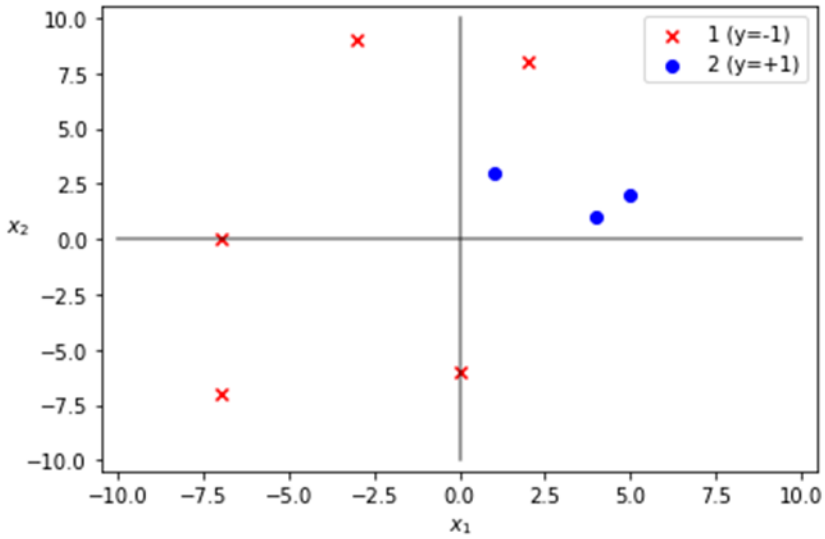

Figure 11: instances for sub-dataset $\mathcal{D}_3$

For this dataset are going to choose another straight line that passes through the origin and has a negative slope. The difference between $f_3(x)$ and $f_1(x)$ is that the line from $f_3(x)$ will have a much higher slope. We will make the line pass through the points (0, 0) and (-1, 10):

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

$$x_2 - 0 = \frac{10 - 0}{-1 - 0} \times (x_1 - 0) \rightarrow x_2 = -10x_1$$

Hence, this weak classifier would have the following form:

$$\boldsymbol{f_3(x) = \begin{cases} +1, & \textit{if } x_2 + 10x_1 \geq 0 \\ -1, & \textit{otherwise} \end{cases}}$$

This weak classifier represented in a 2D plot would look like the following:

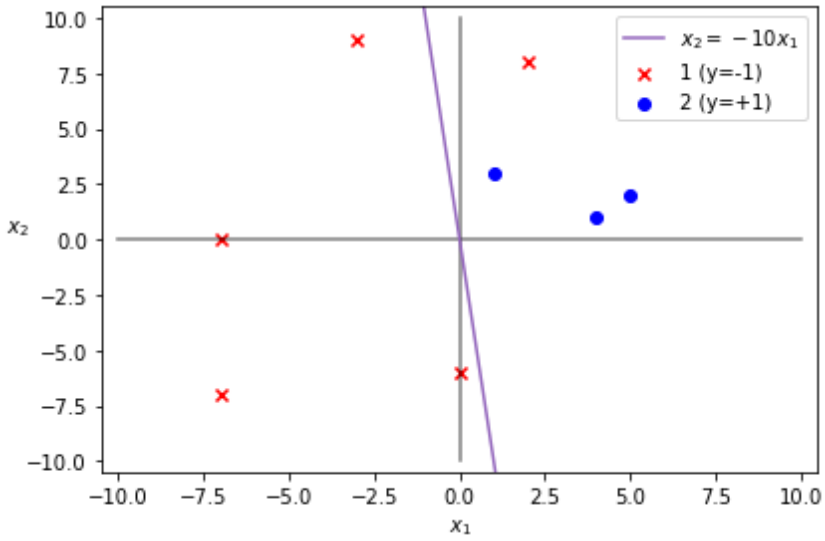

Figure 12: 2D representation of the weak classifier $f_3(x)$

Now, just like before, we will see the accuracy of this classifier to verify that it suits us. We will create a table just like the ones shown before:

INSTANCE	ACTUAL CLASS	OUTPUT OF $f_1(x)$	CORRECT CLASSIFICATION
(-3, 9)	-1	-1	YES
(0, -6)	-1	-1	YES
(-7, 0)	-1	-1	YES
(2, 8)	-1	+1	NO
(-7, -7)	-1	-1	YES
(1, 3)	+1	+1	YES
(5, 2)	+1	+1	YES
(4, 1)	+1	+1	YES

Table 7: Classification error for the third dataset ($\mathcal{D}_3$) and the third weak classifier ($f_3(x)$)

Therefore, the overall accuracy for $f_3(x)$ is:

$$Accuracy\ for\ f_3(x) = \frac{7}{8} \times 100 = 87.5\% > 50\%\ (OK)$$

We can conserve it as its accuracy is above 50%.

We have already finished the first part of the Bagging method. Now we have to apply the designed weak classifiers and bagging classifier to all the samples of our dataset $\mathcal{D}$. We will present the results like asked in the question description, completing a table.

First of all, we must combine all 3 weak classifiers to obtain a final Bagging ensemble model. This means that a classification will be made after interpreting the result of the three weak classifiers. There are different possibilities to classify a given point. One of them is majority voting, as shown in the Appendix 1, another option could be averaging. We will be using the averaging scheme:

$$f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right) \rightarrow f_{\text{final}}(x) = sgn\left(\frac{1}{3} f_1(x) + \frac{1}{3} f_2(x) + \frac{1}{3} f_3(x)\right)$$

This scheme corresponds to an ensemble model with a parallel structure, where all classifiers will make their decision independently. Later, the decision will be combined by the combiner. The combiner can use a plethora of combination strategies (averaging, voting, weighted voting, adaptive weights, etc.) In our specific case, we will be using the averaging scheme, which equation is shown just before.

This parallel structure can be imagined like the following figure from the module slides:

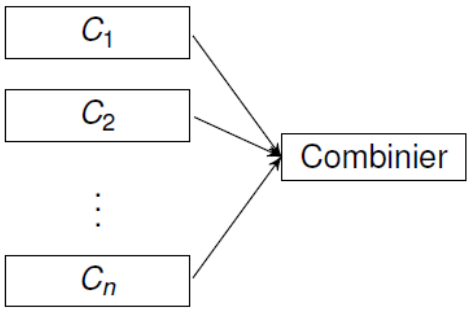

Figure 13: Parallel structure scheme for an ensemble model

Now, we have to complete the table showing the final classification result for each sample. Before that, I will represent the 3 weak classifiers in a 2D plot alongside with the dataset $\mathcal{D}$:

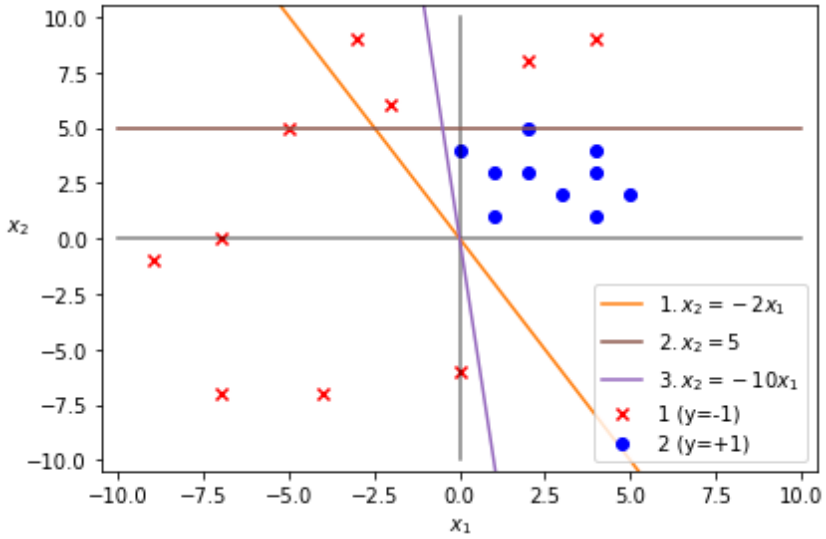

Figure 14: Data instances and the three weak classifiers

Next, I will complete the table shown in the question description. Then, I will try to explain the proceeding followed with a data instance or two as an example. Remember that the overall classifier works with the averaging scheme, meaning that it stays with the sign obtained after summing and averaging the three weak classifiers.

Data (class)	Weak classifier 1	Weak classifier 2	Weak classifier 3	Overall classifier
(-7, 0), -1	-1	+1	-1	-1
(-5, 5), -1	-1	+1	-1	-1
(-4, -7), -1	-1	+1	-1	-1
(-3, 9), -1	+1	-1	-1	-1
(-2, 6), -1	+1	-1	-1	-1
(0, -6), -1	-1	+1	-1	-1
(4, 9), -1	+1	-1	+1	+1
(2, 8), -1	+1	-1	+1	+1
(-7, -7), -1	-1	+1	-1	-1
(-9, -1), -1	-1	+1	-1	-1
(0, 4), +1	+1	+1	+1	+1
(1, 1), +1	+1	+1	+1	+1
(1, 3), +1	+1	+1	+1	+1
(2, 3), +1	+1	+1	+1	+1
(2, 5), +1	+1	+1	+1	+1
(3, 2), +1	+1	+1	+1	+1
(4, 1), +1	+1	+1	+1	+1
(4, 3), +1	+1	+1	+1	+1
(4, 4), +1	+1	+1	+1	+1
(5, 2), +1	+1	+1	+1	+1
Accuracy (%)	80%	70%	90%	90%

Table 8: Classification results using Bagging technique combining 3 weak classifiers

As we can see, the final classifier performs slightly better than most of the weak classifiers. Anyway, there are still very little classifiers to achieve a much better overall accuracy. The next thing we will do before passing to the Bagging ensemble model with 4 weak classifiers will be to explain through a couple of examples how the previous table was achieved.

We will present an example from class 1 and from class 2:

- Data sample (-7, 0) from class 1 ($y = -1$): as we said before, the Bagging ensemble method combines different weak classifiers in parallel. This means that each classifier will output its independent classification result, and then the three outputs will be combined in the combiner (see *Figure 13*) by an averaging scheme. Therefore, the first thing we will have to do is to pass the sample through the three weak classifiers ($f_1(x), f_2(x)$ and $f_3(x)$):

$$\boldsymbol{f_1(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow \text{-7} + 2 \times 0 = \text{-7} \leq 0 \rightarrow \boldsymbol{f_1}((\text{-7}, 0)) = -\mathbf{1}$$

$$\boldsymbol{f_2(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 \leq 5} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow 0 \leq 5 \rightarrow \boldsymbol{f_2}((\text{-7}, 0)) = +\mathbf{1}$$

$$\boldsymbol{f_3(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 10x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow \text{-7} + 10 \times 0 = \text{-70} \leq 0 \rightarrow \boldsymbol{f_3}((\text{-7}, 0)) = -\mathbf{1}$$

As we can see, the first and third weak classifier have outputted a $-\mathbf{1}$, which means it belongs to class 1 (red cross). The second weak classifier, on their behalf, has outputted $+\mathbf{1}$, which means it belongs to class 2 (blue circles). Now we have to pass these three outputs trough the averaging scheme in the combiner:

$$f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right) \rightarrow f_{\text{final}}(x) = sgn\left(\frac{1}{3}f_1(x) + \frac{1}{3}f_2(x) + \frac{1}{3}f_3(x)\right)$$

$$f_{\text{final}}(x) = sgn\left(\frac{1}{3}f_1(x) + \frac{1}{3}f_2(x) + \frac{1}{3}f_3(x)\right) = sgn\left(\frac{1}{3}(-1) + \frac{1}{3}(+1) + \frac{1}{3}(-1)\right)$$

$$\boldsymbol{f_{\text{final}}(x)} = sgn\left(\frac{1}{3}(-1 + 1 - 1)\right) = sgn\left(\frac{-1}{3}\right) = -\mathbf{1}$$

The combiner has outputted the final result, a $-\mathbf{1}$, meaning it belongs to class 1 (red crosses)

- Data sample (1, 1) from class 1 ($y = +1$): we will repeat the process done before. First, we have to pass this sample through the three classifiers to obtain their independent result:

$$\boldsymbol{f_1(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow 1 + 2 \times 1 = 3 \geq 0 \rightarrow \boldsymbol{f_1}((1, 1)) = +\mathbf{1}$$

$$\boldsymbol{f_2(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 \leq 5} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow 1 \leq 5 \rightarrow \boldsymbol{f_2}((1, 1)) = +\mathbf{1}$$

$$\boldsymbol{f_3(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 10x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow 1 + 10 \times 1 = 11 \geq 0 \rightarrow \boldsymbol{f_3}((1, 1)) = +\mathbf{1}$$

As we can see, each independent weak classifier has outputted a $+\mathbf{1}$, which means that each one of them has classified the point as a blue circle from class 2. Averaging the same number gives as a result that same number. In any case, we will pass it through the combiner:

$$\boldsymbol{f_{\text{final}}(x)} = sgn\left(\frac{1}{3}(+1 + 1 + 1)\right) = sgn\left(\frac{+3}{3}\right) = sgn(+1) = +\mathbf{1}$$

We can do this with each and every data sample, but the outcome will be the one shown in *Table 8*. There are only two misclassified samples. Hopefully, as we add weak classifiers to our Bagging model, these misclassifications will be little by little be reduced.

Anyways, by adding a new weak classifier, there will be not much difference. As these misclassified samples will at most have two weak classifiers agreeing that it is from class 1 and the other two agreeing it is from class 2, making a tie.

BAGGING ENSEMBLE MODEL WITH 4 WEAK CLASSIFIERS

We will proceed just like we did before. We have to sample another sub-dataset of 8 instances (STEP 2), then we will train the fourth weak classifier with this new sub-dataset, $\mathcal{D}_4$ (STEP 3). The last step will be to combine this last classifier, $f_4(x)$, to the other 3 weak classifiers to obtain a Bagging ensemble model (STEP 4).

Just like before, we will use the *.sample()* built-in function to create this new sub-dataset. We will reflect it in a table and we will also represent it in a 2D plot.

4. $\mathcal{D}_4$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-7, 7)	(2, 3)
2	(2, 8)	(2, 5)
3	(-2, 6)	(1, 1)
4	(-9, -1)	(4, 3)

Table 9: instances for sub-dataset $\mathcal{D}_4$

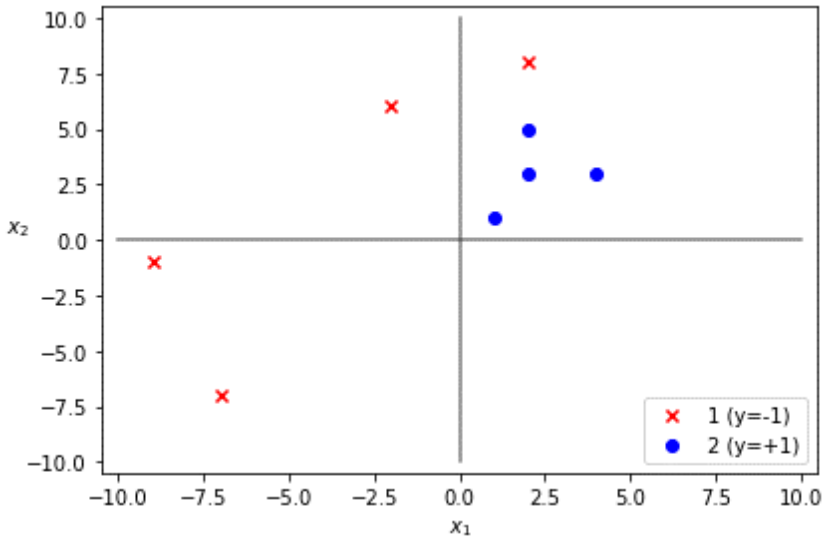

Figure 15: instances for sub-dataset $\mathcal{D}_4$

We could choose as our weak classifier a straight line with positive slope that passes through the origin and between the points (2, 8), from class 1, and (2, 5), from class 2. This way we would have all the points correctly classified. For this, we could use the data points (0,0) and, for example (2, 6), an intermediate point between (2, 8) and (2, 5). Substituting in the equation shown at the beginning, we would have the following:

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

$$x_2 - 0 = \frac{6 - 0}{2 - 0} \times (x_1 - 0) \rightarrow x_2 = 3x_1$$

Therefore, we would have $f_4(x) \equiv x_2 = 3x_1$, where the slope is $m = 3$, and the intercept with the x_2-axis is $b = 0$. This weak classifier represented in a 2-dimensional plot would look something like this:

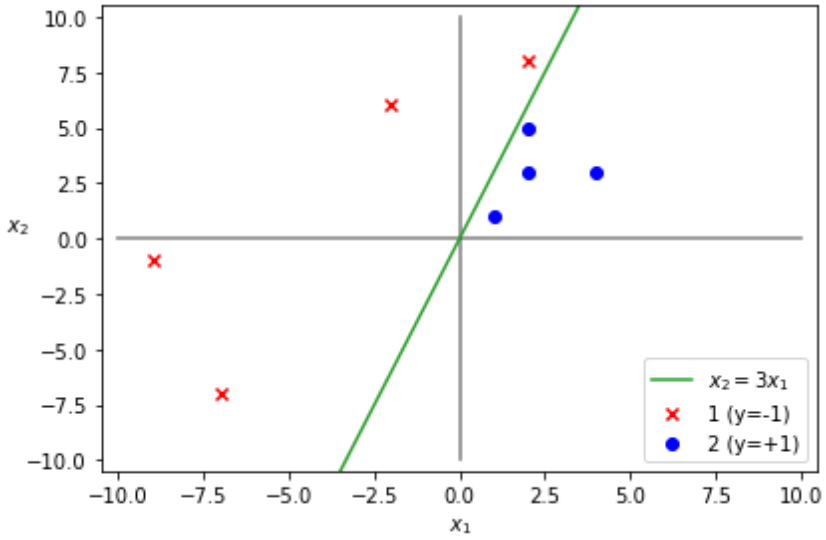

Figure 16: 2D representation of the weak classifier $f_4(x)$

We already have the straight line, now we need the weak classifier per se. This classifier will have the following form:

$$\boldsymbol{f_4(x) = \begin{cases} +1, & \textit{if } x_2 - 3x_1 \leq 0 \\ -1, & \textit{otherwise} \end{cases}}$$

Now we will determine the classification accuracy to check if it is above 50%, if not, we should discard it. As it can be inferred from the *Figure 16*, this weak classifier correctly classifies all points from the sub-dataset $\mathcal{D}_4$. Hence:

$$Accuracy\ for\ f_4(x) = \frac{8}{8} \times 100 = 100\% > 50\%\ (OK)$$

We can conserve it as its accuracy is above 50%.

The next step would be to combine the 4 classifiers we have at hand for the moment. Just like before, we are going to combine these 4 classifiers in parallel, obtaining an independent result for each one of them. After that, we will combine its output through the combiner. This combiner will mainly sum and average the 4 independent results, to give a final output for each data sample.

First, we will plot the 4 weak classifiers alongside with all the data points in our dataset $\mathcal{D}$ to see how it looks in the 2D space:

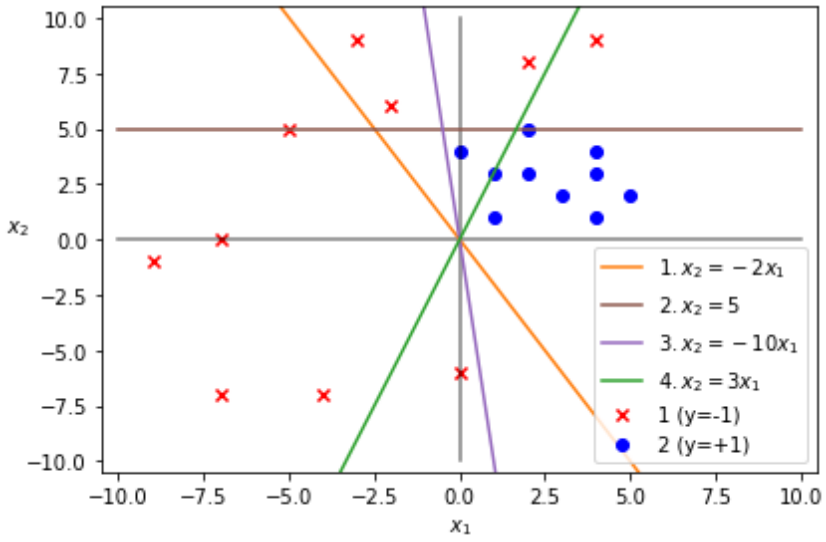

Figure 17: data instances and the four weak classifiers

Next, I will complete the table shown in the question description. Then, I will try to explain the proceeding followed with a data instance as an example. Remember that the overall classifier works with the averaging scheme, meaning that it stays with the sign obtained after summing and averaging the four weak classifiers. As there are 4 classifiers, there will be ties. We will leave the ties as unclassified samples, meaning they are also misclassified samples as the class is not returned. The table is the following:

Data (class)	Weak classifier 1	Weak classifier 2	Weak classifier 3	Weak classifier 4	Overall classifier
(-7, 0), -1	-1	+1	-1	-1	-1
(-5, 5), -1	-1	+1	-1	-1	-1
(-4, -7), -1	-1	+1	-1	-1	-1
(-3, 9), -1	+1	-1	-1	-1	-1
(-2, 6), -1	+1	-1	-1	-1	-1
(0, -6), -1	-1	+1	-1	+1	??
(4, 9), -1	+1	-1	+1	+1	+1
(2, 8), -1	+1	-1	+1	-1	??
(-7, -7), -1	-1	+1	-1	-1	-1
(-9, -1), -1	-1	+1	-1	-1	-1
(0, 4), +1	+1	+1	+1	-1	+1
(1, 1), +1	+1	+1	+1	+1	+1
(1, 3), +1	+1	+1	+1	+1	+1
(2, 3), +1	+1	+1	+1	+1	+1
(2, 5), +1	+1	+1	+1	+1	+1
(3, 2), +1	+1	+1	+1	+1	+1
(4, 1), +1	+1	+1	+1	+1	+1
(4, 3), +1	+1	+1	+1	+1	+1
(4, 4), +1	+1	+1	+1	+1	+1
(5, 2), +1	+1	+1	+1	+1	+1
Accuracy (%)	80%	70%	90%	85%	85%

Table 10: Classification results using Bagging technique combining 4 weak classifiers

As we can see from the previous table, the accuracy has somewhat decreased, a 5%. This is not very explanatory because now, instead of having 2 misclassified instances, we have only one. The other two samples have an undefined class, they are unclassified.

Looking carefully, the two unclassified instances belong to class 1. Maybe if we set the rule that every sample that has a tie as the final classification should belong to class 1. Anyway, just like before, I am going to explain how we obtained the results from *Table 10* by showing an example. Say we choose an instance with a tie:

- Data sample (0, -6) from class 1 ($y = -1$): we will repeat the process done before. First, we have to pass this sample through the three classifiers to obtain their independent result:

$$\boldsymbol{f_1(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ (-6) + 2 \times 0 = -6 \leq 0 \rightarrow\ \boldsymbol{f_1}((0,-6)) = -\mathbf{1}$$

$$\boldsymbol{f_2(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 \leq 5} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ (-6) \leq 5 \rightarrow\ \boldsymbol{f_2}((0,-6)) = +\mathbf{1}$$

$$\boldsymbol{f_3(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 10x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ (-6) + 10 \times 0 = -6 \leq 0 \rightarrow\ \boldsymbol{f_3}((0,-6)) = -\mathbf{1}$$

$$\boldsymbol{f_4(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 - 3x_1 \leq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ (-6) - 3 \times 0 = -6 \leq 0 \rightarrow\ \boldsymbol{f_4}((0,-6)) = +\mathbf{1}$$

As we can see, each independent weak classifier has outputted either a $+\mathbf{1}$, or a $-\mathbf{1}$. This means that they are not at all in accordance. Averaging these four outputs we end up with the following final classification after passing it through the combiner:

$$f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right) \rightarrow f_{\text{final}}(x) = sgn\left(\frac{1}{4}f_1(x) + \frac{1}{4}f_2(x) + \frac{1}{4}f_3(x) + \frac{1}{4}f_4(x)\right)$$

$$f_{\text{final}}(x) = sgn\left(\frac{1}{4}(-1) + \frac{1}{4}(+1) + \frac{1}{4}(-1) + \frac{1}{4}(+1)\right)$$

$$\boldsymbol{f_{\text{final}}(x)} = sgn\left(\frac{1}{4}(-1 + 1 - 1 + 1)\right) = sgn\left(\frac{0}{4}\right) = ??$$

We can see that the sign of zero is undefined, therefore so is the class of this instance. We could handle different possibilities to solve this issue. One of them is saying that the sign of zero is -1 for this exercise. Other option is to average the result with the raw output of the weak classifiers. Because, as we know from the SVM module, the raw output of $\boldsymbol{f_i(x)}$ somehow represents the distance of the point to that straight line.

For the time being, I am going to leave these instances as misclassified. Nonetheless, I would have taken the first option presented in the last paragraph: attributing the class 1 to the ties. This way we would have an overall classification accuracy of 95%, only misclassifying one point.

For the next Bagging ensemble model, we may break the ties in the unclassified instances, as we will have an odd number of weak classifiers.

BAGGING ENSEMBLE MODEL WITH 5 WEAK CLASSIFIERS

We will proceed just like we did before. We have to sample another sub-dataset of 8 instances (STEP 2), then we will train the fifth weak classifier with this new sub-dataset, $\mathcal{D}_5$ (STEP 3). The

last step will be to combine this last classifier, $f_5(x)$, to the previous 4 weak classifiers to obtain a Bagging ensemble model (STEP 4).

Just like before, we will use the *.sample()* built-in function to create this new sub-dataset. We will reflect it in a table like the ones shown before, and we will also represent it in a 2D plot. This is the STEP 2:

1. $\mathcal{D}_4$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(2, 8)	(4, 1)
2	(4, 9)	(3, 2)
3	(-3, 9)	(5, 2)
4	(0, -6)	N/A
5	(-7, -7)	N/A

Table 11: instances for sub-dataset $\mathcal{D}_5$

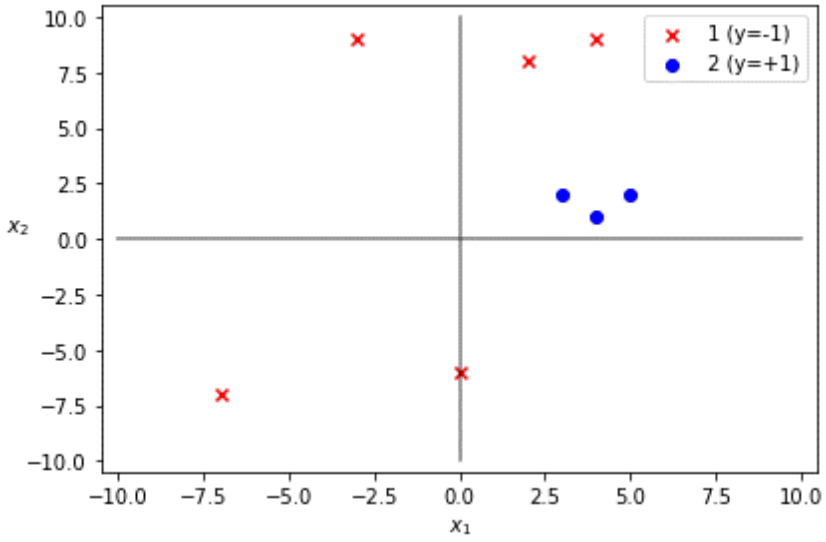

Figure 18: instances for sub-dataset $\mathcal{D}_5$

As we can see from the previous table (*Table 11*) and figure (*Figure 18*), the function *.sample()* has returned 5 instances from class 1 (red crosses) and 3 instances from class 2 (blue circles), just like with the sub-dataset $\mathcal{D}_5$.

Now, we must continue with STEP 3, where we train a new weak classifier, $f_5(x)$, with this random sub-dataset. We will do just like before, choose two points from the 2D space and obtain the straight line that passes through them.

Observing the data, we may correctly classify all instances by passing a line between the red crosses at (4, 9) and (0, -6), and the blue circle at (4, 1). First, let us see if this line accurately divides the instances from one class from the other. For this purpose, we must choose two points in the 2-dimensional space to obtain a line from them. We will choose two points one unit under the red crosses, this is: (4, 8) and (0, -7). Substituting these points in the equation we obtain the following:

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

$$x_2 - 8 = \frac{(-7) - 8}{0 - 4} \times (x_1 - 4) \rightarrow x_2 = \frac{15x_1}{4} - 7$$

Therefore, we would have $f_4(x) \equiv x_2 = \frac{15x_1}{4} - 7$, where the slope is $m = \frac{15}{4}$, and the intercept with the x_2-axis is $b = -7$, just below the point (0, -6). This weak classifier represented in a 2-dimensional plot would look something like this:

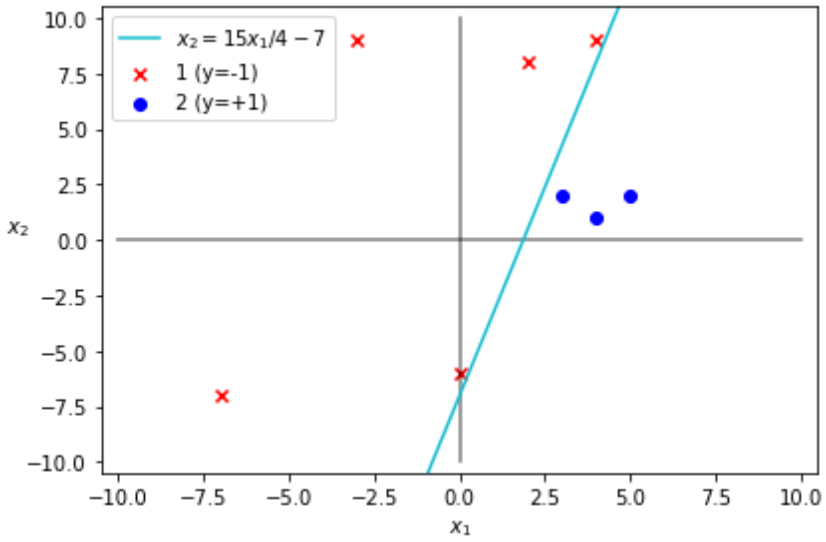

Figure 19: 2D representation of the weak classifier $f_5(x)$

As we can see, this weak classifier classifies accurately the 8 samples in the sub-dataset, meaning it is feasible. We already have the straight line, now we need the weak classifier per se. This classifier will have the following form:

$$\boldsymbol{f_5(x) = \begin{cases} +1, & \textit{if } x_2 - \frac{15x_1}{4} + 7 \leq 0 \\ -1, & \textit{otherwise} \end{cases}}$$

The next step would be to combine the 5 classifiers we have at hand for the moment (STEP 4). Just like before, we are going to combine these classifiers in parallel, obtaining an independent result for each one of them. After that, we will combine its output through the combiner. This combiner will mainly sum and average the 5 independent results, to give a final output for each data sample.

First, we will plot the 5 weak classifiers alongside with all the data points in our dataset $\mathcal{D}$ to see how it looks in the 2D space:

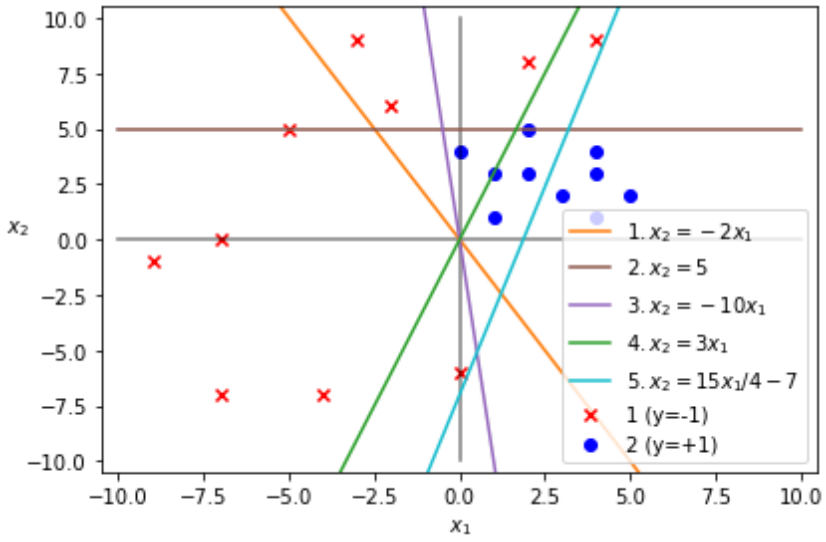

Figure 20: data instances and the five weak classifiers

Next, I will complete the table shown in the question description. Then, I will try to explain the proceeding followed with a data instance as an example. Remember that the overall classifier works with the averaging scheme, meaning that it stays with the sign obtained after summing and averaging the five weak classifiers. The table is the following:

Data (class)	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	**Overall classifier**
(-7, 0), -1	-1	+1	-1	-1	-1	-1
(-5, 5), -1	-1	+1	-1	-1	-1	-1
(-4, -7), -1	-1	+1	-1	-1	-1	-1
(-3, 9), -1	+1	-1	-1	-1	-1	-1
(-2, 6), -1	+1	-1	-1	-1	-1	-1
(0, -6), -1	-1	+1	-1	+1	-1	-1
(4, 9), -1	+1	-1	+1	+1	-1	+1
(2, 8), -1	+1	-1	+1	-1	-1	-1
(-7, -7), -1	-1	+1	-1	-1	-1	-1
(-9, -1), -1	-1	+1	-1	-1	-1	-1
(0, 4), +1	+1	+1	+1	-1	-1	+1
(1, 1), +1	+1	+1	+1	+1	-1	+1
(1, 3), +1	+1	+1	+1	+1	-1	+1
(2, 3), +1	+1	+1	+1	+1	-1	+1
(2, 5), +1	+1	+1	+1	+1	-1	+1
(3, 2), +1	+1	+1	+1	+1	+1	+1
(4, 1), +1	+1	+1	+1	+1	+1	+1
(4, 3), +1	+1	+1	+1	+1	+1	+1
(4, 4), +1	+1	+1	+1	+1	+1	+1
(5, 2), +1	+1	+1	+1	+1	+1	+1
Accuracy (%)	80%	70%	90%	85%	75%	95%

Table 12: Classification results using Bagging technique combining 5 weak classifiers

The naming of the columns has changed a bit, but they still represent the same thing: the output of each weak classifier.

As we can see from the previous table, the accuracy has increased, classifying almost every point correctly. The actual overall accuracy is 95%, almost perfect. Currently, we only have one data instance incorrectly classified, this would be the example (4, 9).

Now I will explain the procedure followed to obtain the previous table by presenting an example from the dataset. For this case, we will choose (2, 8):

- Data sample (2, 8) from class 1 ($y = -1$): we will repeat the process done before. First, we have to pass this sample through the three classifiers to obtain their independent result:

$$\boldsymbol{f_1}(\boldsymbol{x}) = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 8 + 2 \times 2 = 12 \geq 0 \rightarrow\ \boldsymbol{f_1}((2,8)) = +\mathbf{1}$$

$$\boldsymbol{f_2}(\boldsymbol{x}) = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 \leq 5} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 8 \geq 5 \rightarrow\ \boldsymbol{f_2}((2,8)) = -\mathbf{1}$$

$$\boldsymbol{f_3}(\boldsymbol{x}) = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 10x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 8 + 10 \times 2 = 28 \geq 0 \rightarrow\ \boldsymbol{f_3}((2,8)) = +\mathbf{1}$$

$$\boldsymbol{f_4}(\boldsymbol{x}) = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 - 3x_1 \leq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 8 - 3 \times 2 = 2 \geq 0 \rightarrow\ \boldsymbol{f_4}((2,8)) = -\mathbf{1}$$

$$\boldsymbol{f_5}(\boldsymbol{x}) = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 - \frac{15x_1}{4} + 7 \leq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 8 - \frac{15 \times 2}{4} + 7 = \frac{15}{2} \geq 0 \rightarrow\ \boldsymbol{f_5}((2,8)) = -\mathbf{1}$$

As we can see, each independent weak classifier has outputted either a $+\mathbf{1}$, or a $-\mathbf{1}$. This means that they are not at all in accordance. Averaging these five outputs we end up with the following final result after passing it through the combiner:

$$f_{\text{final}}(x) = sgn\left(\sum_{i=1}^{M} \frac{1}{M} f_i(x)\right)$$

$$f_{\text{final}}(x) = sgn\left(\frac{1}{5}f_1(x) + \frac{1}{5}f_2(x) + \frac{1}{5}f_3(x) + \frac{1}{5}f_4(x) + \frac{1}{5}f_5(x)\right)$$

$$f_{\text{final}}(x) = sgn\left(\frac{1}{5}(+1) + \frac{1}{5}(-1) + \frac{1}{5}(+1) + \frac{1}{5}(-1) + \frac{1}{5}(-1)\right)$$

$$\boldsymbol{f_{\text{final}}}(\boldsymbol{x}) = sgn\left(\frac{1}{5}(1 - 1 + 1 - 1 - 1)\right) = sgn\left(\frac{-1}{5}\right) = -\mathbf{1}$$

Finally, we end up with a final classification that is correct. This point belongs to class 1 because its overall output is negative.

When observed the *Figure 20*, we can infer that the space is divided in regions. These regions are the spaces between the weak classifiers. For example, we will use as an example the region at the left between the weak classifiers $\boldsymbol{f_1}(\boldsymbol{x})$ (orange), $\boldsymbol{f_2}(\boldsymbol{x})$ (brown) and $\boldsymbol{f_4}(\boldsymbol{x})$ (green). This regions contains some points of our dataset like (-7, 0) or (-4, -7), among others.

Well, if we focus on these region, every point contained within its boundaries will have exactly the same output for each weak classifier, obtaining the same output for the overall Bagging ensemble model. In fact, by looking at the outputs of the points contained in this region ((-9, -1), (-7, 0), (-7, -7) and (-4, -7)) reflected in *Table 12*, we observe that all their outputs are exactly the same.

As all the weak classifiers output the same result, the overall classifier will also output the same result. This is logical as we are dividing the space in different regions. Each region will be characterised by a set of outputs, and by averaging or majority voting those outputs, each region will also be characterised by a final output from the overall classifier.

We could check the last statement by looking at some points in the same region, like for example the points located in (-3, 9) and in (-2, 6), which belong to class 1 ($y = -1$). These points find themselves enclosed between three classifiers: $\boldsymbol{f_1(x)}$ (orange), $\boldsymbol{f_2(x)}$ (brown) and $\boldsymbol{f_3(x)}$ (purple). If we now focus on the *Table 12* and the independent outputs they had for each weak classifier, we observe that all of them have the same results:

- $\boldsymbol{f_1(x)}$ outputs $+\mathbf{1}$,
- $\boldsymbol{f_2(x)}$ outputs $-\mathbf{1}$,
- $\boldsymbol{f_3(x)}$ outputs $-\mathbf{1}$,
- $\boldsymbol{f_4(x)}$ outputs $-\mathbf{1}$, and
- $\boldsymbol{f_5(x)}$ outputs $-\mathbf{1}$.

After combining these 5 weak classifiers, we have an overall classification of $-\mathbf{1}$, meaning it belongs to class 1. These could be done with every and each one of the regions created by these classifiers.

We can infer that referring to the number of weak classifiers, the more the better, unless we end up with an even number of them. By having an even number of weak classifiers we will have the problem of the ties. Yet it is true that this problem can be avoided by agreeing on a policy like the one I defined before: in case there is a tie, the instance is classified as class 1, for example. There are other possibilities like averaging the raw output too.

Regardless of the number of models used to form the Bagging classifier, the more the better. It will allow to create more regions in our 2D space, enabling more possibilities of classification. Two points really close to each other may represent different classes because they belong to two different regions.

At the end, having 17 weak classifiers will give us a better performance than having 5 in the great majority of cases. We could find some exceptions where the sampling of the sub-datasets has been a bit weird, not sampling some key data samples or sampling too many times some other points. This is an intrinsic condition of the Bagging methodology.

Q4. Design a Boosting classifier consisting of 3 weak classifiers using the steps shown in Appendix 2. A linear classifier should be used as a weak classifier. Explain and show the design of the hyperplanes of weak classifiers. List the parameters of the design hyperplanes. After designing the weak classifiers, apply the designed weak classifiers and boosting classifier to all the samples in Table 1. Present the classification results in a table as shown in Table 2. Explain how to determine the class (for each weak classifier and boosting classifier) using one test sample. Comment on the results of the overall classifier in terms of classification performance when comparing with the 1st, 2nd and the 3rd weak classifiers, and with the bagging classifier with 3-weak classifiers in Q.3. (30 Marks)

The Boosting method is another type of ensemble model. It is somewhat similar to the Bagging method, but it has some differences. This type of ensemble model are created by first generating a classifier with accuracy on the training set greater than average. Then, new components are added to form an ensemble whose joint decision rule has arbitrarily high accuracy on the training set. So, we are going to create informative datasets to create an ensemble classifier. Hence, the classifiers are more dependent on each other than in the Bagging method.

We will start by designing the weak classifiers of this Boosting method. The procedure to carry out a Boosting ensemble model is presented in the Appendix 2 of the Coursework description:

- STEP 1: start with dataset $\mathcal{D}$ with n patterns. We have our dataset defined in Q1 and plotted in Q2.
- STEP 2: training procedure:
    - STEP 2.1: randomly select a set of $n_1 \leq n$ patterns (without replacement) from $\mathcal{D}$ to create dataset $\mathcal{D}_1$. Train a weak classifier C_1 using $\mathcal{D}_1$ (C_1 should have at least 50% classification accuracy).
    - STEP 2.2: create an "informative" dataset $\mathcal{D}_2$ $(n_2 \leq n)$ from $\mathcal{D}$ of which roughly half of the patterns should be correctly classified by C_1 and the rest is wrongly classified. Train a weak classifier C_2 using $\mathcal{D}_2$.
    - STEP 2.3: create an "informative" dataset $\mathcal{D}_3$ from $\mathcal{D}$ of which the patterns are not well classified by C_1 and C_2 (C_1 and C_2 disagree). Train a weak classifier C_3 using $\mathcal{D}_3$.
- STEP 3: the final decision of classification is based on the votes of the weak classifiers, e.g., by the first two weak classifiers if they agree, and by the third weak classifier if the first two disagree.

The key goal of this method is to improve the accuracy of any given learning algorithm. In our case, as the question description demands us to create linear classifiers, we want to improve the accuracy of each one of them as we add more weak classifiers. It could be said that the classification performance is "boosted" by combining a number of weak classifiers (having accuracy only slightly better than chance is sufficient).

Therefore, we have to add weak classifiers one by one to form the overall classifier such that higher classification accuracy can be achieved. Each weak classifier is trained with "informative" dataset complementary to other. This makes each sub-dataset dependent on the previous ones. Thereby, each weak classifier is also dependent on the previous ones.

Once we know the steps we must follow, we can begin creating the Boosting classifier. We have to create an ensemble model of 3 weak classifiers. Therefore, we will first create 3 weak

classifiers by the procedure explained before. We will begin by the first weak classifier, $\boldsymbol{C_1(x)}$, which will be drawn from the first sub-dataset, $\mathcal{D}_1$. As this sampling has nothing to do with the one done in Q3, we will start from scratch.

First of all, we have to decide how many patterns will our first sub-dataset, $\mathcal{D}_1$. We have to choose a number of samples, and the unique requirement is that it has to be smaller or equal than n, the number of training samples, which in our case is 20. We will select n_1 to be 8, like before.

We will select these instances using the *.sample()* built-in function from Python. Like in Q3, I will create a table with these instances and right after, I will plot these instances to see how they are represented in the 2D space.

1. $\mathcal{D}_1$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-4, -7)	(3, 2)
2	(-5, 5)	(4, 1)
3	(-7, 0)	(4, 4)
4	(-9, -1)	(2, 5)

Table 13: instances for sub-dataset $\mathcal{D}_1$

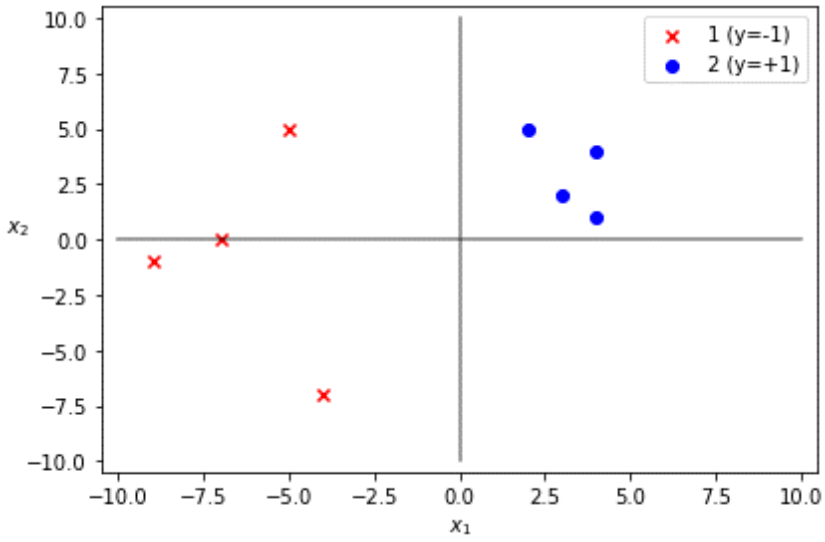

Figure 21: instances for sub-dataset $\mathcal{D}_1$

The next step is to train a weak classifier, $\boldsymbol{C_1(x)}$, with this sub-dataset. We will proceed just like we did in Q3, creating a linear classifier "by inspection", this is observing the dataset and finding a straight line that would perform good, or at least better than a random one.

Looking to the data, a linear classifier like the one used in Q3 ($\boldsymbol{f_1(x)}$) could work. I am talking about the straight line with slope $m = -2$ that passes through the origin. This classifier would perform perfectly on this sub-dataset, achieving a 100% of accuracy. As we already know how this line is obtained, I will not repeat the process. The first weak classifier for the Boosting method is the following:

$$\boldsymbol{C_1(x)} = \begin{cases} \boldsymbol{+1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ \boldsymbol{-1}, & \boldsymbol{otherwise} \end{cases}$$

And this linear classifier represented in the 2D space would look like this with $\mathcal{D}_1$:

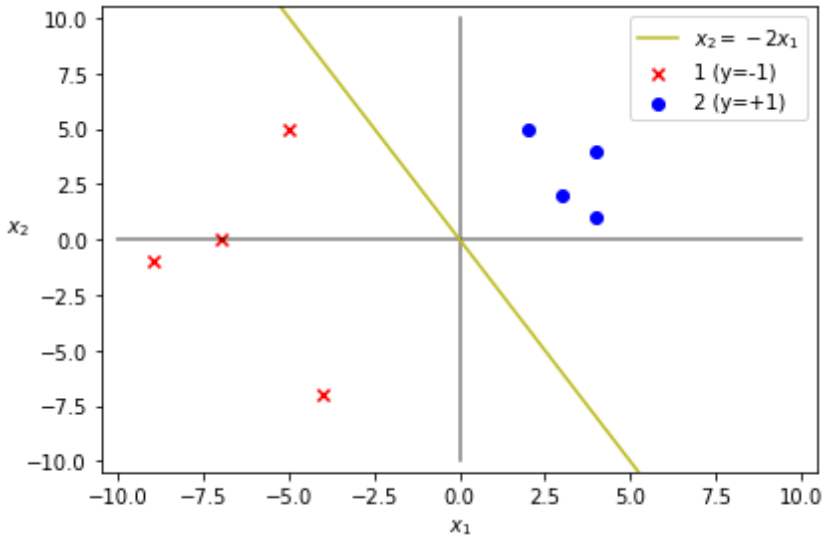

Figure 22: 2D representation of the weak classifier $C_1(x)$

There is no need to check the accuracy of this classifier as it can be inferred visually: the instances at the top right corner will be classified as class 2 $(y = +1)$; on the other hand, the instances in the bottom left corner will be classified as class 1 $(y = -1)$. This classifier has 100% of accuracy for the sub-dataset $\mathcal{D}_1$. Hence, this classifier suits us well and we do not need no other.

The next step would be STEP 2.2, we have to resample our dataset. First of all, we have to remove the instances in $\mathcal{D}_1$ used for $\boldsymbol{C_1(x)}$. Then, from the remaining instances in $\mathcal{D}$, we have to select a number of them $(n_2 \leq n)$. This sub-dataset, $\mathcal{D}_2$, must have one condition, roughly half of its patterns should be correctly classified by $\boldsymbol{C_1(x)}$, and the rest should be wrongly classified.

This means we need to use $\boldsymbol{C_1(x)}$ to determine whether the samples from $\mathcal{D}_2$ can be handled by $\boldsymbol{C_2(x)}$. This is what we meant before when we stated that each weak classifier is dependent on the previous ones. That is why we also call these sub-datasets "informative", because $\mathcal{D}_2$ depends on $\mathcal{D}_1$, thereby, on the classification performance of $\boldsymbol{C_1(x)}$ too.

Consequently, we will sample 6 instances from the remaining 12 instances in our original dataset $(n_2 = 6)$. After that, we will check if it satisfies the condition stated before, we will pass them through the first weak classifier, $\boldsymbol{C_1(x)}$ and see how well they are the classified.

We will start by sampling and plotting this second sub-dataset, but before that, we will plot the remaining instances after discarding the ones included in $\mathcal{D}_1$, and we will call this sub-dataset $\mathcal{D}_{remaining}$:

1.1. $\mathcal{D}_{remaining}$

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(-3, 9)	(0, 4)
2	(-2, 6)	(1, 3)
3	(0, -6)	(1, 1)
4	(4, 9)	(2, 3)
5	(2, 8)	(4, 3)
6	(-7, 7)	(5, 2)

Table 14: instances for sub-dataset $\mathcal{D}_{remaining}$

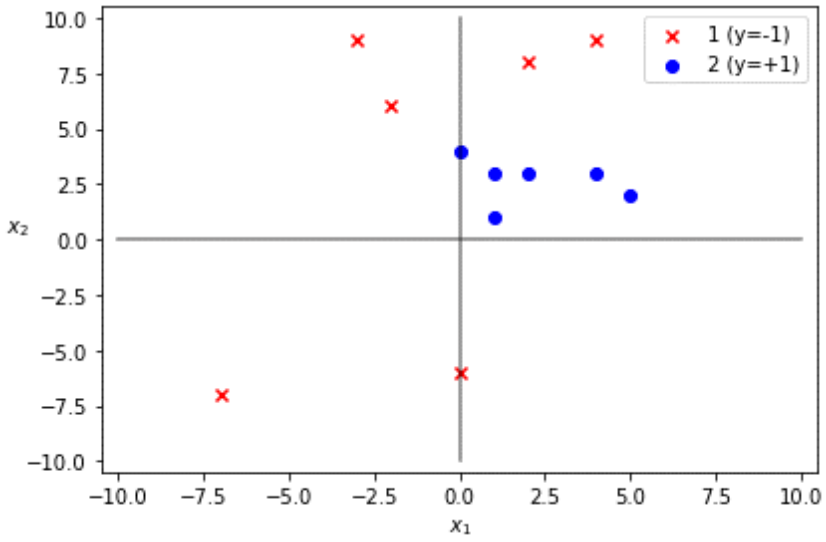

Figure 23: instances for sub-dataset $\mathcal{D}_{remaining}$

Now, we have to sample a new sub-dataset, $\mathcal{D}_2$, from this remaining dataset $\mathcal{D}_{remaining}$. We will do like before, benefitting from the built-in function *.sample()*. We will show the 6 instances from this dataset in a table, and then we will plot them in the 2D space to see where they are located:

2. $\mathcal{D}_2$: as we can see, this sub-dataset has more instances from class 2 than from class 1. It does not matter, we will carry on and see if these instances satisfy the condition and have half of them misclassified by the first weak classifier, $\boldsymbol{C_1(x)}$.

INSTANCE	CLASS 1: $y = -1$	CLASS 2: $y = +1$
1	(2, 8)	(5, 2)
2	(4, 9)	(1, 3)
3	N/A	(4, 3)
4	N/A	(1, 1)

Table 15: instances for sub-dataset $\mathcal{D}_2$

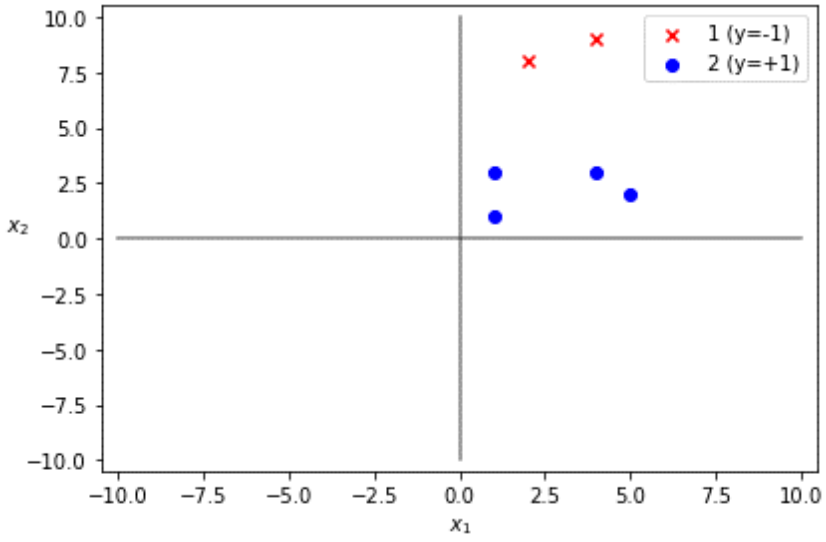

Figure 24: instances for sub-dataset $\mathcal{D}_2$

Now we have to check if these instances satisfy the condition mentioned before. We will pass them through $\boldsymbol{C_1(x)}$ and see how many are correctly classified and how many are misclassified:

INSTANCE	ACTUAL CLASS	PREDICTED CLASS BY $\boldsymbol{C_1(x)}$	CORRECT CLASSIFICATION
(2, 8)	-1	+1	NO
(4, 9)	-1	+1	NO
(5, 2)	+1	+1	YES
(1, 3)	+1	+1	YES
(4, 3)	+1	+1	YES
(1, 1)	+1	+1	YES

Table 16: Classification error for $\mathcal{D}_2$ and the first weak classifier $\mathrm{C}_1(\mathrm{x})$

Here we will put the process followed for predicting the class of these instances. We will set an example with the first instance, (2, 8). The only thing we have to do is pass this instances through the classifier:

$$\boldsymbol{C_1(x)} = \begin{cases} \boldsymbol{+1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ \boldsymbol{-1}, & \boldsymbol{otherwise} \end{cases} \rightarrow 8 + 2 \times 2 = 12 \geq 0 \rightarrow \boldsymbol{f_1((2,8)) = +1}$$

As we can see, this instance is erroneously classified by this classifier. We can also interpret the *Table 16* by observing the following figure. The instances at the upper right side of the line are classified as $+1$. On the other hand, the instances located in the opposite region will always be classified as -1.

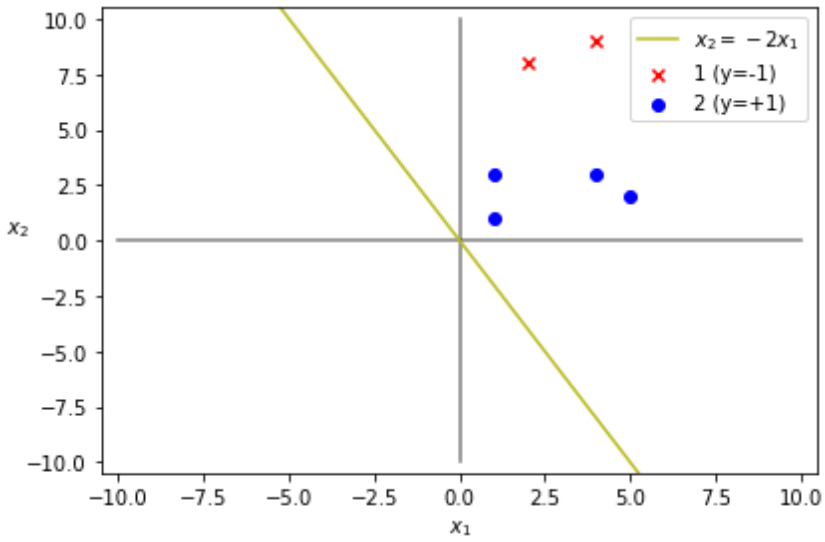

Figure 25: representation of the weak classifier $C_1(x)$ with the sub-dataset $\mathcal{D}_2$

Now, by seeing the *Table 16* and *Figure 25* we can decide if we can use this classifier or not. The accuracy is:

$$Accuracy\ for\ C_1(x) = \frac{4}{6} \times 100 = 66.67\% > 50\%\ (OK)$$

As we can see, this dataset suits us for the Boosting method because it has roughly half the instances correctly classified . The next step is to train a weak classifier with this dataset. We will do it "by observation" again, by passing a straight line through two points in between both classes, say we choose the points to be $x_1 = (1, 5)$ and $x_2 = (4, 7)$.

From the equation used before in Q3 we have the following:

$$x_2 - x_{2,1} = \frac{x_{2,2} - x_{2,1}}{x_{1,2} - x_{1,1}} \times (x_1 - x_{1,1})$$

$$x_2 - 5 = \frac{7\text{-}5}{4\text{-}1} \times (x_1 - 1) \rightarrow x_2 = \frac{2}{3}x_1 + \frac{13}{3}$$

Therefore, we would have $C_2(x) \equiv x_2 = \frac{2}{3}x_1 + \frac{13}{3}$, where the slope is $m = \frac{2}{3}$, and the intercept with the x_2-axis is $b = \frac{13}{3}$, just above the point (0, 4). This weak classifier with the sub-dataset $\mathcal{D}_2$ represented in a 2-dimensional plot would look something like this:

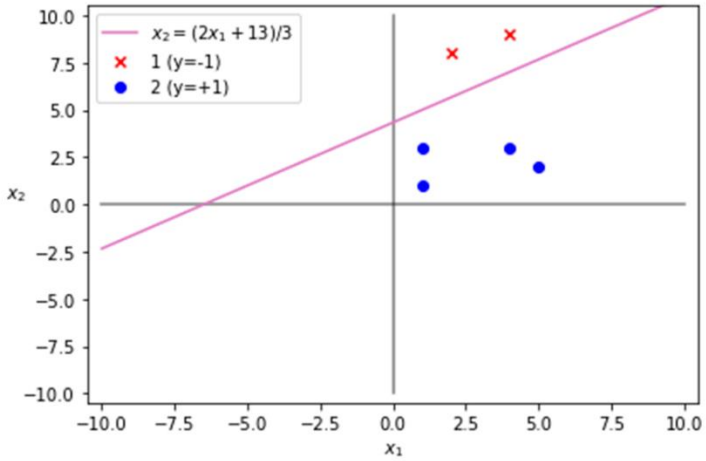

Figure 26: 2D representation of the weak classifier $C_2(x)$

And the equation of this weak classifier is the following:

$$\boldsymbol{C_2(x) = \begin{cases} +1, & \textit{if } x_2 - \frac{2}{3}x_1 - \frac{13}{3} \leq 0 \\ -1, & \textit{otherwise} \end{cases}}$$

There is no need to check the accuracy of this classifier as it can be inferred visually: the points in the lower right corner, the blue circles, are all classified as class 2 ($y = +1$); and the ones above the pink line, the red crosses, are all classified as class 1 ($y = -1$). Hence, this classifier has 100% of accuracy for the sub-dataset $\mathcal{D}_1$, enough because it has better performance than a random one. Therefore, this classifier suits us well and we do not need no other.

$$Accuracy\ for\ C_2(x) = \frac{6}{6} \times 100 = 100\% > 50\%\ (OK)$$

The next step would be STEP 2.3. Lastly, we have to create a third sub-dataset, $\mathcal{D}_3$, from the remaining instances of the original dataset without $\mathcal{D}_1$, $\mathcal{D}_{remaining}$. These instances have to be misclassified either by $\boldsymbol{C_1(x)}$ or by $\boldsymbol{C_2(x)}$. Meaning that they cannot agree on the class of that instance. We are now going to plot the instances from $\mathcal{D}_{remaining}$ that satisfy that condition:

INSTANCE	CORRECT PREDICTION BY $\boldsymbol{C_1(x)}$?	CORRECT PREDICTION BY $\boldsymbol{C_2(x)}$?
(-3, 9)	NO	YES
(-2, 6)	NO	YES
(0, -6)	YES	NO
(4, 9)	NO	YES
(2, 8)	NO	YES
(-7, 7)	YES	NO

Table 17: class prediction by $C_1(x)$ and $C_2(x)$for the sub-dataset $\mathcal{D}_3$

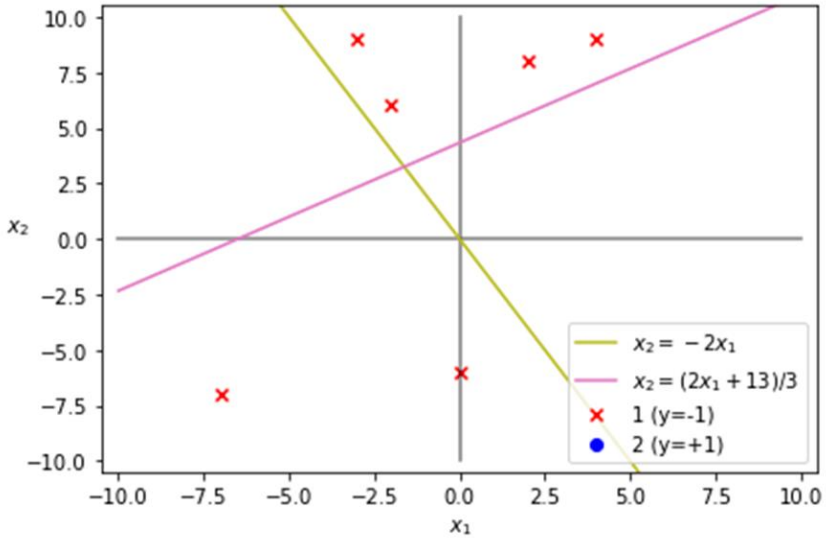

Figure 27: 2D representation of the samples misclassified by $C_1(x)$ and $C_2(x)$

If we divide the previous plot in 4 regions we end up with two of them agreeing (the regions at left and right side), and two of them disagreeing (the upper and lower region). The rest of data points are the blue circles in the right region, the instances from class 2 that are all correctly classified; and some samples from class 1 that are correctly classified too because they lay in the left region.

The last step is to train a linear classifier model from this dataset, which we will call $\boldsymbol{C_3(x)}$. As we can see, all of them belong to the same class, the class 1. Hence, there is not actually any model really needed, as if the two classifiers do not agree, it means that the data instance is from class 1 $(y = -1)$. Anyways, we will draw a linear classifier "by inspection". We will create a straight line parallel to the vertical axis that intercepts the x_1-axis in $x_1 = 10$, for example.

In the 2-dimensional space it will look like the following:

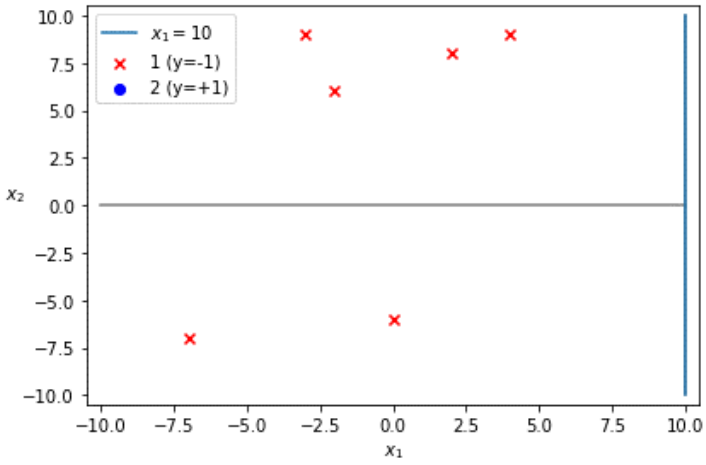

Figure 28: 2D representation of the weak classifier $C_3(x)$ with the sub-dataset $\mathcal{D}_3$

This line will have the following classifying equation:

$$\boldsymbol{C_3(x)} = \begin{cases} \boldsymbol{+1}, & \boldsymbol{if\ x_1 \geq 10} \\ \boldsymbol{-1}, & \boldsymbol{otherwise} \end{cases}$$

Meaning that if a data sample is at the left side of the blue line in $x_1 = 10$, the line will be classified as a red cross from class 1 ($y = -1$).

We already have our Boosting ensemble model prepared. Now we will check how well it performs by applying this designed model to all the samples in $\mathcal{D}$. To correctly evaluate the performance of this classifier we have to have in mind that the final decision of classification is based on the votes of the weak classifiers.

We will complete the table shown in Q3 (see *Table 8*) with these classifiers. We will put the output of each weak classifier and then the overall classification depending on the output of the previous three. This time if the first two classifiers ($\boldsymbol{C_1(x)}$ and $\boldsymbol{C_2(x)}$) agree we attribute the class they predict. In case they disagree, we look at the third classifier ($\boldsymbol{C_3(x)}$) and attribute the class this last classifier predicts, which in our case will always be class 1.

Data (class)	Weak classifier 1 ($\boldsymbol{C_1(x)}$)	Weak classifier 2 ($\boldsymbol{C_2(x)}$)	Weak classifier 3 ($\boldsymbol{C_3(x)}$)	Overall classifier
(-7, 0), -1	-1	-1	N/A	-1
(-5, 5), -1	-1	-1	N/A	-1
(-4, -7), -1	-1	+1	-1	-1
(-3, 9), -1	+1	-1	-1	-1
(-2, 6), -1	+1	-1	-1	-1
(0, -6), -1	-1	+1	-1	-1
(4, 9), -1	+1	-1	-1	-1
(2, 8), -1	+1	-1	-1	-1
(-7, -7), -1	-1	+1	-1	-1
(-9, -1), -1	-1	-1	N/A	-1
(0, 4), +1	-1	-1	N/A	+1
(1, 1), +1	-1	-1	N/A	+1
(1, 3), +1	-1	-1	N/A	+1
(2, 3), +1	-1	-1	N/A	+1
(2, 5), +1	-1	-1	N/A	+1
(3, 2), +1	-1	-1	N/A	+1
(4, 1), +1	-1	-1	N/A	+1
(4, 3), +1	-1	-1	N/A	+1
(4, 4), +1	-1	-1	N/A	+1
(5, 2), +1	-1	-1	N/A	+1
Accuracy (%)	80%	85%	100%	100%

Table 18: Classification results using Boosting technique combining 3 weak classifiers

This Boosting ensemble model performs perfectly, achieving an accuracy of 100%. We could resume the last table like the following: if we divide the 2D space with the two first weak classifiers ($\boldsymbol{C_1(x)}$ and $\boldsymbol{C_2(x)}$) in four regions, the region of the right will classify the instances as class 2 ($y = +1$); and the other three regions (the upper, lower and left region) will classify any instance that is there as class 1 ($y = -1$).

We can check the last statement in the following figure:

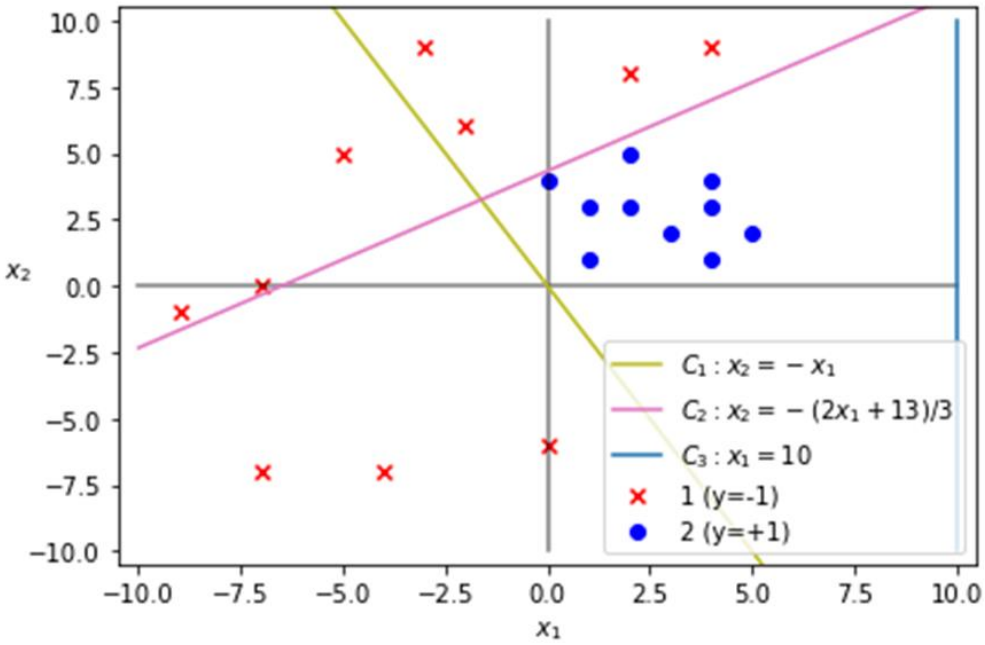

Figure 29: 2D representation of the samples of the initial dataset $\mathcal{D}$ with the three weak classifiers

We will show an example of the procedure we have followed with each data instance with an example. We will use the data point (4, 9) from class 1 $(y = -1)$:

- Data sample (4, 9) from class 1 $(y = -1)$: we will repeat the process done before. To begin, we have to pass this sample through the two first classifiers to obtain their independent result:

$$\boldsymbol{C_1(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 + 2x_1 \geq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 9 + 2 \times 4 = 17 \geq 0 \rightarrow\ \boldsymbol{C_1((4,9))} = +\mathbf{1}$$

$$\boldsymbol{C_2(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_2 - \frac{2}{3}x_1 - \frac{13}{3} \leq 0} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 9 - \frac{2}{3}4 - \frac{13}{3} = 2 \geq 0 \rightarrow\ \boldsymbol{C_2((4,9))} = -\mathbf{1}$$

As this two classifiers disagree, we have to pass it through the last one, $\boldsymbol{C_3(x)}$, to predict its final class:

$$\boldsymbol{C_3(x)} = \begin{cases} +\mathbf{1}, & \boldsymbol{if\ x_1 \geq 10} \\ -\mathbf{1}, & \boldsymbol{otherwise} \end{cases} \rightarrow\ 4 \leq 10 \rightarrow\ \boldsymbol{C_3((4,9))} = -\mathbf{1}$$

From this last result, we can assure that the final prediction is -1 as the third and last classifier predicted this class.

$$\boldsymbol{f_{\text{final}}(x)} = -\mathbf{1}$$

FINAL CONCLUSION

This Boosting classifier performs perfectly, it achieves an accuracy of 100%. As I explained before, it has divided the space quite accurately and isolated the blue points from class 2 really accurately. This model only works because the three weak classifiers work correctly and in concordance. If we use them separately, they do not achieve such a good performance.

We have performed two different ensemble models, one by applying the Bagging method (Q3), and another by using the Boosting method (Q4). The one that classified better all data instances was the Boosting ensemble model, it achieved an accuracy of 100%. When comparing it to the Bagging model with three weak classifiers, the one exposed in *Table 8* and *Figure 14*, we see that a couple of instances were erroneously classified because of the averaging scheme. As this method does not use that scheme, it worked better.