Assignment 1

This document contains answers to the "theoretical" questions in the assignment. Please look at this file in combination with the code for more clarity.

1.2:

Looking at the data set we have a clear separation between the two classes (A and B).

For this linear classification problem there are an infinite number of possible boundaries (as long as they separate the data set it is a solution to the problem). All three lines (red, green and blue) are possible solutions to the problem. They are however, probably not the optimal one we are looking for.

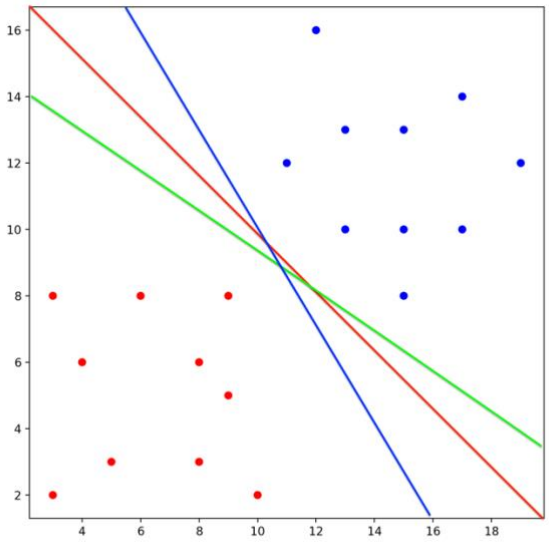

Figure 1 Three suggestions for linear decision boundaries for our classification problem

1.2a

As for which of my boundaries I think would generalize the best I would suggest the red line, as this is the line which has the biggest distance between both parts of the dataset (maximizes the margin between A and B). Although note that I don't think that the red line *actually* maximizes the margin.

1.3

Looking at the two attributes we see that both classes have a lot of data points with the value eight as its AttrY (see Figure 2). This means that separating the two classes based on this attribute will cause problems (atleast if we base the model on this set only) This makes the attribute hard to justify.

If we look at AttrX instead we see a more clear distinction between the classes when looking at our scatter plot, unlike AttrY the classes have no overlapping values. This makes this attribute more fit for distinguishing the two classes.

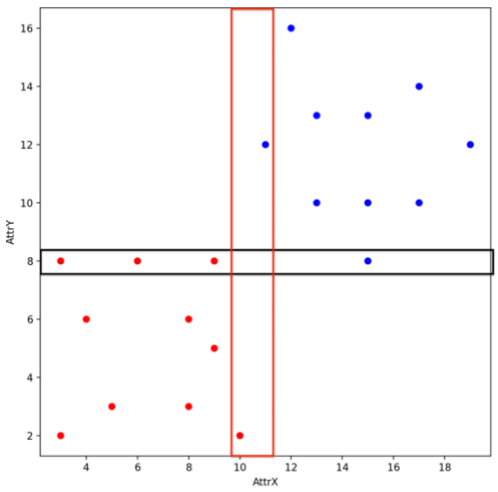

Figure 2 Overlapping AttrY values (Black Box), no overlapping values (Red Box)

To test this hypothesis I decided to use the following attribute evaluation methods in Weka: GainRatioAttributeEval,

InfoGainAttributeEval and OneAttributeEval. I tested them with and without cross validation to see if it impacted the results. By using the a1_training.csv dataset I got the following (I only discuss the final result, the complete evaluation output can be found in the hand-in folder:

Evaluation results

First I used the GainRatioAttributeEval method for evaluation. For this classification method we got a ranking favoring AttrX, this supports my claim which also favored this attribute. The scores for the test were:

- AttrX: 1
- AttrY: 0.764

We see a clear favoring of AttrX however the difference between the two attributes were small. Adding cross validation to the evaluation didn´t change the ranking but it did raise the value of AttrY slightly (0.764 -> 0.771). I suspect however that the lack of impact is because of the size of the dataset. As with the above method InfoGainAttributeEval method also favors AttrX although for this method AttrY scored lower (0.758) further supporting my claim. Lastly, in the OneRAttributeEval the difference between AttrX and AttrY was miniscule (AttrX: 100 and AttrY 95).

2.1

Looking at our chosen decision boundary we get the following results: Point 1 has the values 10, 8 if we look at my decision boundary it should belong to the A class (red). The same goes for points 2 and 4 (they have values (5,5 and 10,4). The Third point should be class B (blue) by my decision boundary, it has the values 17,12.

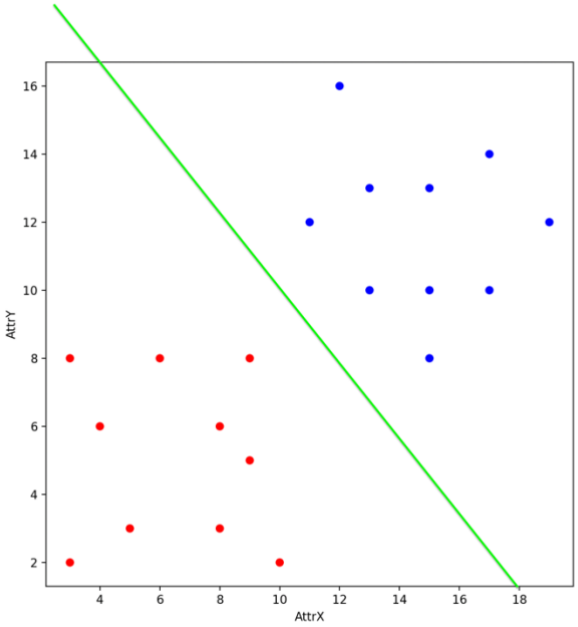

Figure 3 Chosen decision boundary

Using my boundary we get 3 correct classifications (2 A´s and 1 B) and one incorrect classification (Predicted A, actual B)

We get the following confusion matrix:

- TA = True A class (True positive)
- TB = True B class (True negative)
- FA = False A class (False positive)
- FB = False B class (False negative)

Class	A	B	
A	TA = 2	FA = 1	3
B	FB = 0	TB = 1	1
	2	2	

2.1a

To calculate our accuracy we use the following formula with our confusion matrix:

$$Accuracy = \frac{True\ Positive + True\ negative}{Total\ population}$$

In our case we get the following accuracy:

$$Accuracy = \frac{3+1}{4} = 0.75$$

$$\underline{75\%\ accuracy}$$

2.2

Looking at the different classification algorithms we get the exact same confusion matrix and classification accuracy for all of them (see folder wekaFiles for the full output). They have varying error rates however the TP rate, FP Rate and accuracy is the same! I suspect this is because of the size of the datasets though.

Confusion matrix and classification results for IBk, J48 Decision tree, Logistic Reggression and NaïveBayes:

=== Confusion Matrix ===

```
 a b   <-- classified as
 2 0 | a = A
 1 1 | b = B
```

Correctly Classified Instances	3	75	%
Incorrectly Classified Instances	1	25	%

2.3

Applying the J48 Decision tree algorithm to our train and test data set we get the tree shown I figure 4. The tree is very simple and will classify the points based on attribute X in the following way:

- If the AttrX value is above or equal to 10 its classified as A
- If the AttrX value is below 10 its classified as B.

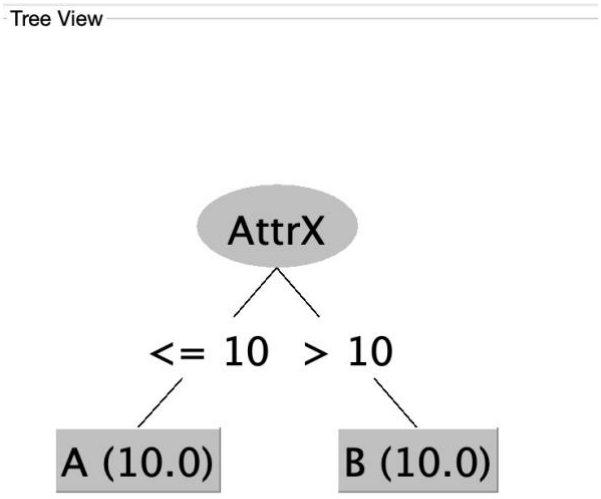

Figure 4 Decision tree

2.4

By increasing the dataset the algorithms I tested with may produce more complicated models which have more intricate ways to discriminate the two classes.

If we use the decision tree as and example it might be able to produce a tree with 3-4 levels and more leafnodes/decisions with a bigger

dataset. Right now the tree is way too simple largely due to the lack of training data. By having a bigger dataset we will be able to more effectively use methods like cross validation to improve our models. The biggest benefit to increasing the dataset is that we will (hopefully) find better features to distinguish the two classes. More specifically having more data generally means smaller error margins.