Saúl Abimael Gómez Martínez A01700790

Lab 7: Decision trees

Part 1:

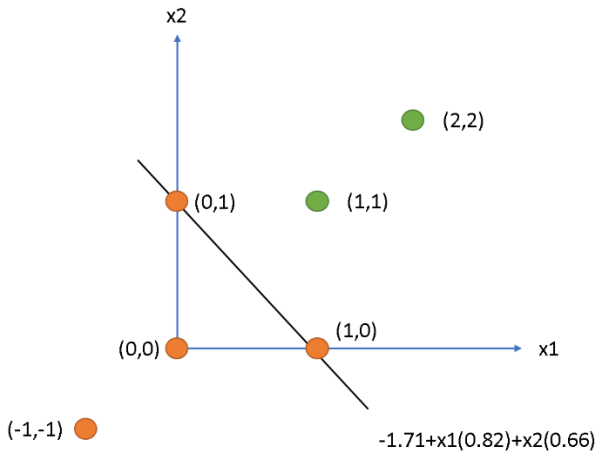
For plotting the linearly separable example I used the test for an OR function. The following values are the input variables like dimensionality, number of training Set, number of testing Set, the training Sets and the testing Sets.

```
2
4
3
1,1,1
0,1,1
1,0,1
0,0,0
-1,-1
2,2
0,0
```

The following image show the results of the training Set which are correct or at least are the expected values. The last line shows the weights' values which are going to be used for creating the plot.

```
0
1
0
-1.710 0.820 0.660
```

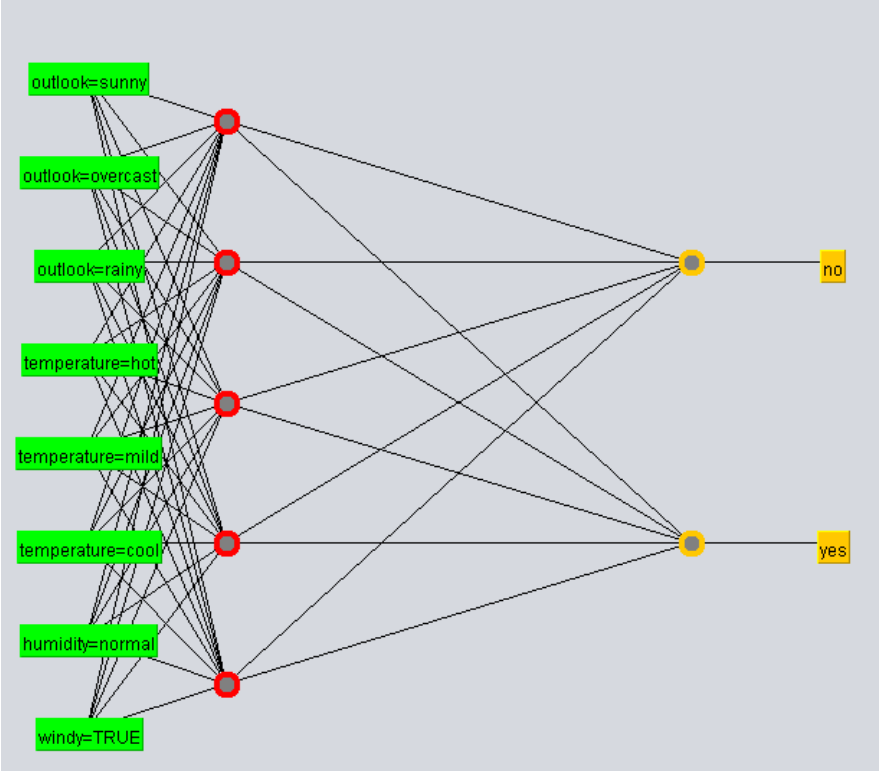
The following plot shows how the values are distributed and the type of result we would expect from it. The green dot represents a positive value while the orange dot is for a negative value. As it can be seen, the function separates almost correctly the data since two dots should be positive, but they aren't, meaning the function requires some fixing. Still manages to separate some values as expected.



Part 2:

The following images show an example of a multilayer perceptron done in WEKA as well as its variations. It was made using the condition for deciding when to play in a day depending of different variables.

The next image shows the first network made, as it can be seen there are a lot of interconnections as well as a lot of inputs. For the case were the inputs are characters it values all its options and not just numbers.



The following tables shows the parameters used and how they are organized.

outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes

overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

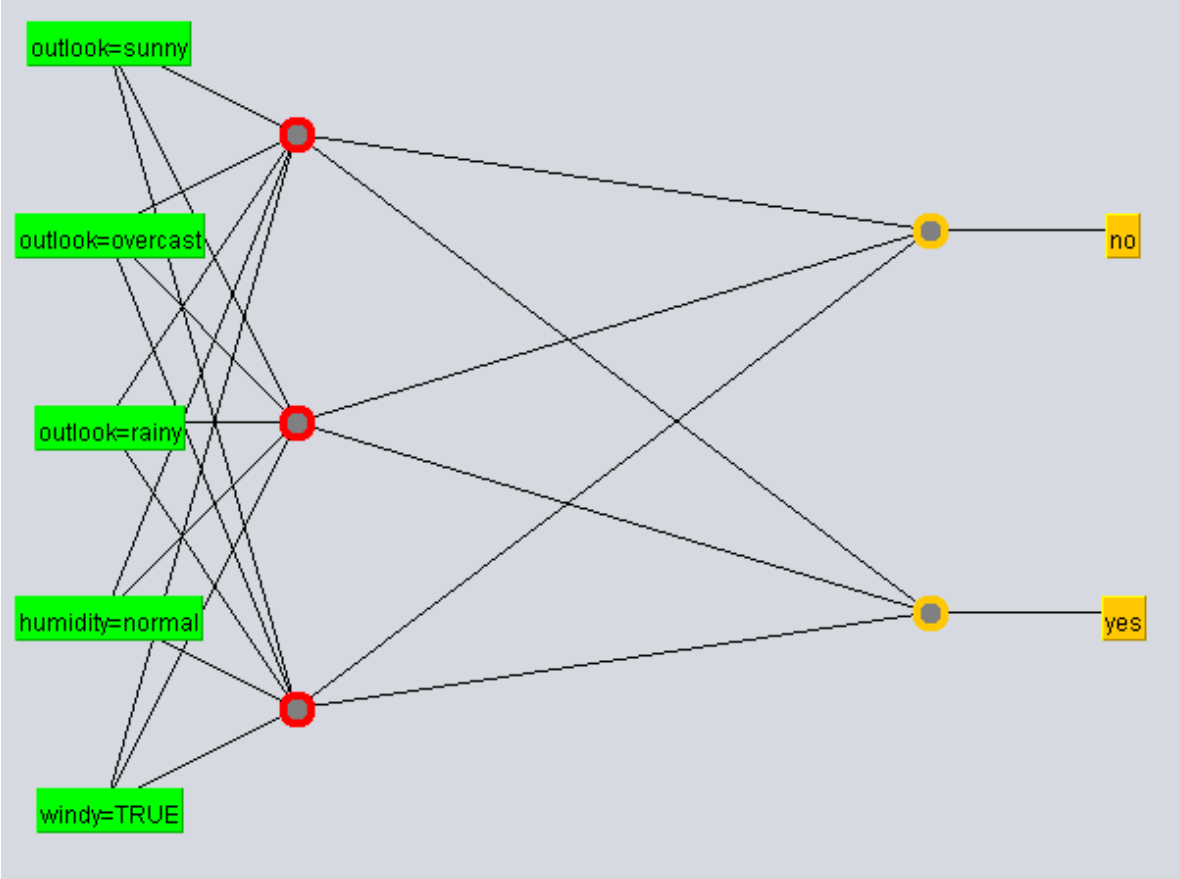
The next image shows the time needed for doing the training which in this case was of 0.01 seconds. The relative absolute error is about 5.35%.

```
Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0248
Root mean squared error                  0.0356
Relative absolute error                  5.3522 %
Root relative squared error              7.4329 %
Total Number of Instances               14
```

For the second network, the connections look like the following image, were the number of nodes has decreased since one parameter has disappeared, which makes sense.



The following tables shows the parameters used and how they are organized.

outlook	humidity	windy	play
sunny	high	FALSE	no
sunny	high	TRUE	no
overcast	high	FALSE	yes
rainy	high	FALSE	yes
rainy	normal	FALSE	yes
rainy	normal	TRUE	no
overcast	normal	TRUE	yes
sunny	high	FALSE	no
sunny	normal	FALSE	yes
rainy	normal	FALSE	yes
sunny	normal	TRUE	yes
overcast	high	TRUE	yes
overcast	normal	FALSE	yes
rainy	high	TRUE	no

In this case the time was reduce to almost 0 meaning it didn't take a lot of time to do the training. From this, its safe to assume that while the number of parameters is proportional to the time of training. In the other hand the error percentage has increased to 7.721% which means that it is inverse proportional to the number of parameters.

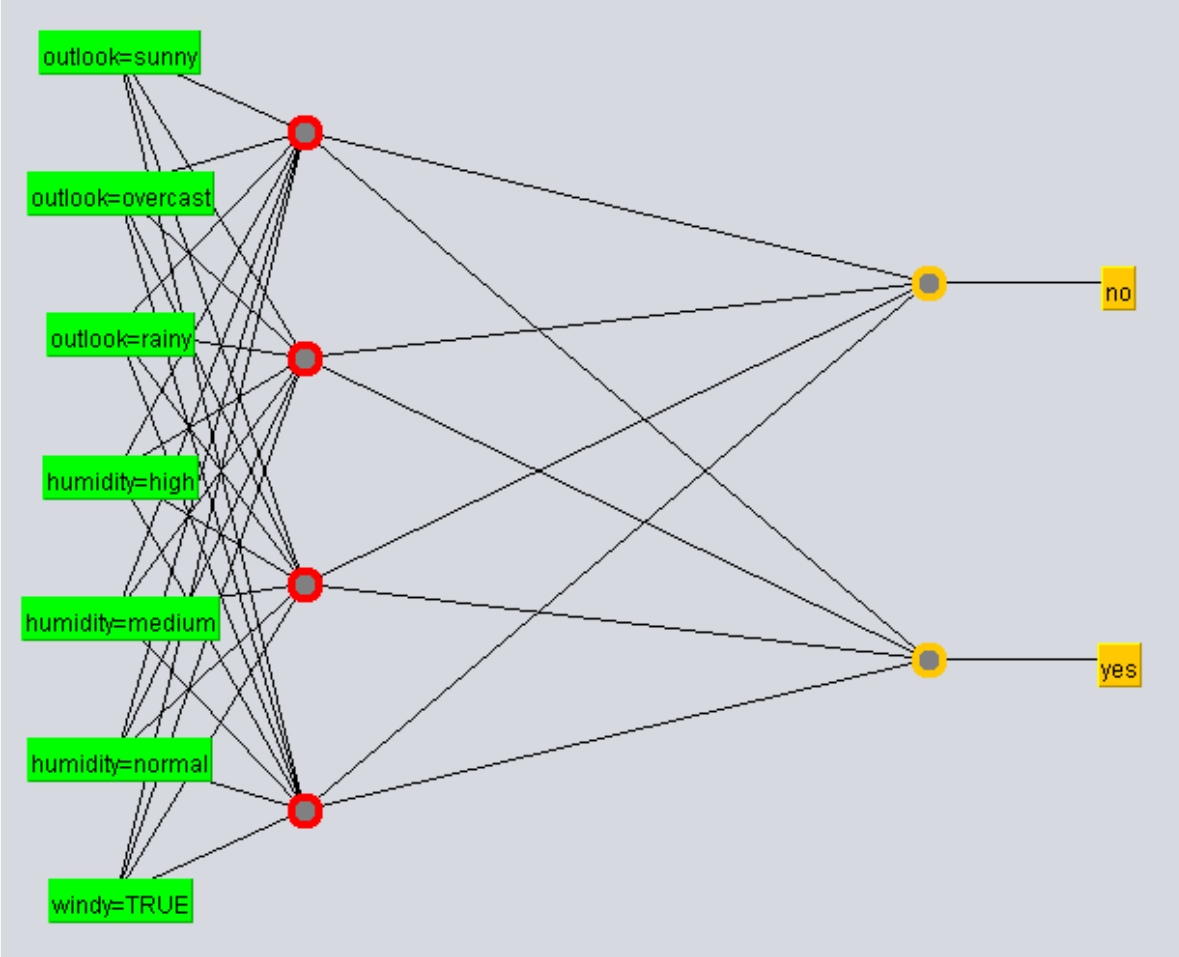
```
Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0358
Root mean squared error                  0.0438
Relative absolute error                  7.7209 %
Root relative squared error              9.1257 %
Total Number of Instances               14
```

From the last information, it can be observed the relations between these variables. The error may work like that due to its data mining being reduced. If the parameters decrease then the data mining will be lower.

The last case looks very similar to the first case, but the difference is that one parameter disappeared and another parameter (humidity) increased one option. One thing to be highlighted about the network is that if the parameter has only two options then it will have only one input but if there are more the number of inputs is the same as the number of options.



The following tables shows the parameters used and how they are organized.

outlook	humidity	windy	play
sunny	high	FALSE	no
sunny	medium	TRUE	no
overcast	medium	FALSE	yes
rainy	high	FALSE	yes
rainy	normal	FALSE	yes
rainy	medium	TRUE	no
overcast	normal	TRUE	yes
sunny	high	FALSE	no
sunny	normal	FALSE	yes
rainy	medium	FALSE	yes
sunny	normal	TRUE	yes

overcast	high	TRUE	yes
overcast	normal	FALSE	yes
rainy	high	TRUE	no

Finally, the time for the training data was the same as the first one but the main difference is that the error was lower than the first one as well, its value was of 4.64%.

```
Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances          14               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0216
Root mean squared error                  0.027
Relative absolute error                  4.6468 %
Root relative squared error              5.6351 %
Total Number of Instances               14
```

It could be assumed that the percentage was lower because the even though the number of parameters was lower than before it still had more conditions to consider so it allowed a deeper data mining.

From this laboratory, it could be learned that ANNs are very useful since they are capable of learning and model non-linear and complex relationships, as well as generalize the information obtained with the intention of making predictions.

As said before one of its best usages is for predicting results so some application where it would be useful are forecasting were a lot of variables need to be analyzed like weather, date, previous data, etc.; also, image processing for inferring complex and nonlinear relationships that an image may have like its pixels.

From its abilities, and advantages they are worth the effort since they allow to solve complex problems which by other means could be much more difficult or less accurate.

The only problem it has is that it cannot solve problems were the values or inputs are very random or unpredictable, it may lead to an infinite loop where it would never be able to give a right answer.