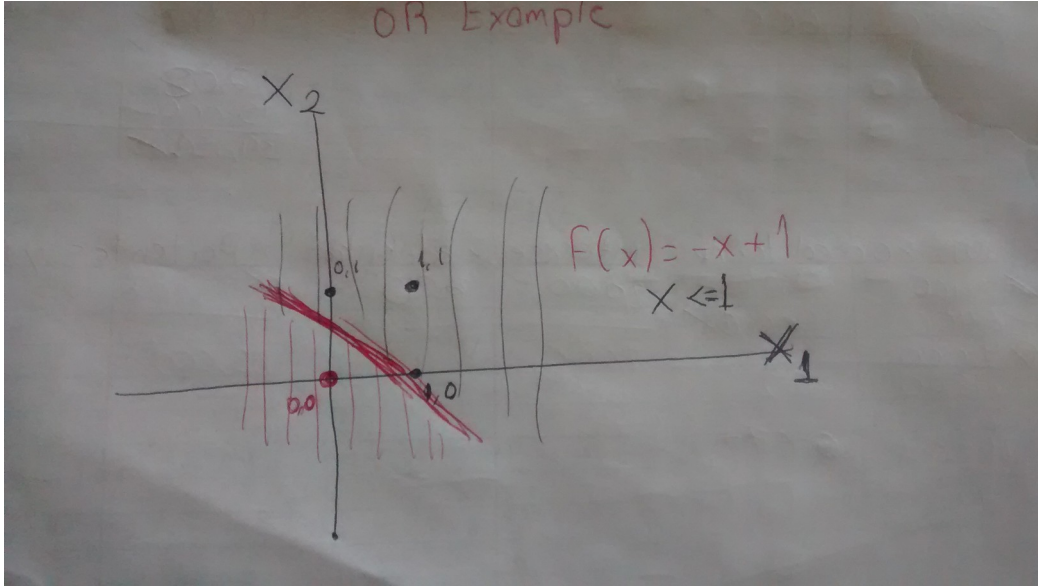


# REPORT

## PART 1



## PART 2

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

18:32:49 - functions.MultilayerPerceptron

Classifier output

Input  
Node 0  
Class no  
Input  
Node 1

Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	11	78.5714 %
Incorrectly Classified Instances	3	21.4286 %
Kappa statistic	0.5116	
Mean absolute error	0.265	
Root mean squared error	0.4627	
Relative absolute error	55.6497 %	
Root relative squared error	93.7923 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.889	0.400	0.800	0.889	0.842	0.519	0.733	0.857	yes
	0.600	0.111	0.750	0.600	0.667	0.519	0.733	0.589	no
Weighted Avg.	0.786	0.297	0.782	0.786	0.779	0.519	0.733	0.761	

=== Confusion Matrix ===

a b <- classified as

8	1	a = yes
2	3	b = no

Status

OK Log x 0

Weka GUI Chooser

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose MultilayerPerceptron

Test options

☒ Use training set  
☐ Supplied test set  
☐ Cross-validation Folds  
☐ Percentage split %  
More options...

(Nom) play

Start

Result list (right-click for options)

18:32:49 - functions.MultilayerPerceptron  
18:35:18 - functions.MultilayerPerceptron  
18:35:40 - functions.MultilayerPerceptron

Status

Building model on training data...

Neural Network

Controls

Start Epoch 0  
Accept Num Of Epochs 500  
Error per Epoch = 0

Learning Rate = 0.3  
Momentum = 0.2

Log x1

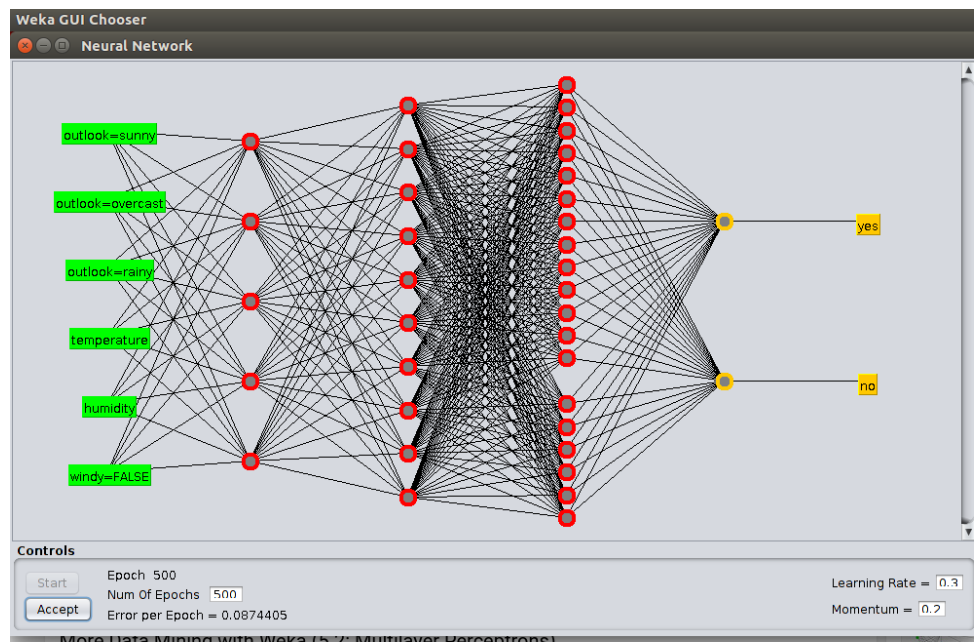
Weka GUI Chooser

Neural Network

Controls

Start Epoch 500  
Accept Num Of Epochs 500  
Error per Epoch = 0.0020689

Learning Rate = 0.3  
Momentum = 0.2



- **Explanations as to what are ANNs good for.**
  - Perhaps the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism that 'learns' from observed data.
  - The advantages of deep neural networks are record-breaking accuracy on a whole range of problems including image and sound recognition, text and time series analysis, etc.
- **Where would you use them?**
  - Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
  - Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
  - Data processing, including filtering, clustering, blind source separation and compression.
  - Robotics, including directing manipulators, prosthesis.
  - Control, including computer numerical control.
- **Are they worth the effort implementing or not?**
  - It depends of the problem, there are other algorithms that are faster but ANNs have a great accuraccy.
- **What kinds of problems do they not solve?**
  - They can be hard to tune to ensure they learn well, and therefore hard to debug;

- They do not have explanatory power; i.e. they mainly extract the best signals to accurately classify and cluster data, but they will not tell you why they reached a certain conclusion;
- They are computationally intensive to train; i.e. you need a lot of chips and a distributed run-time to train on very large datasets.