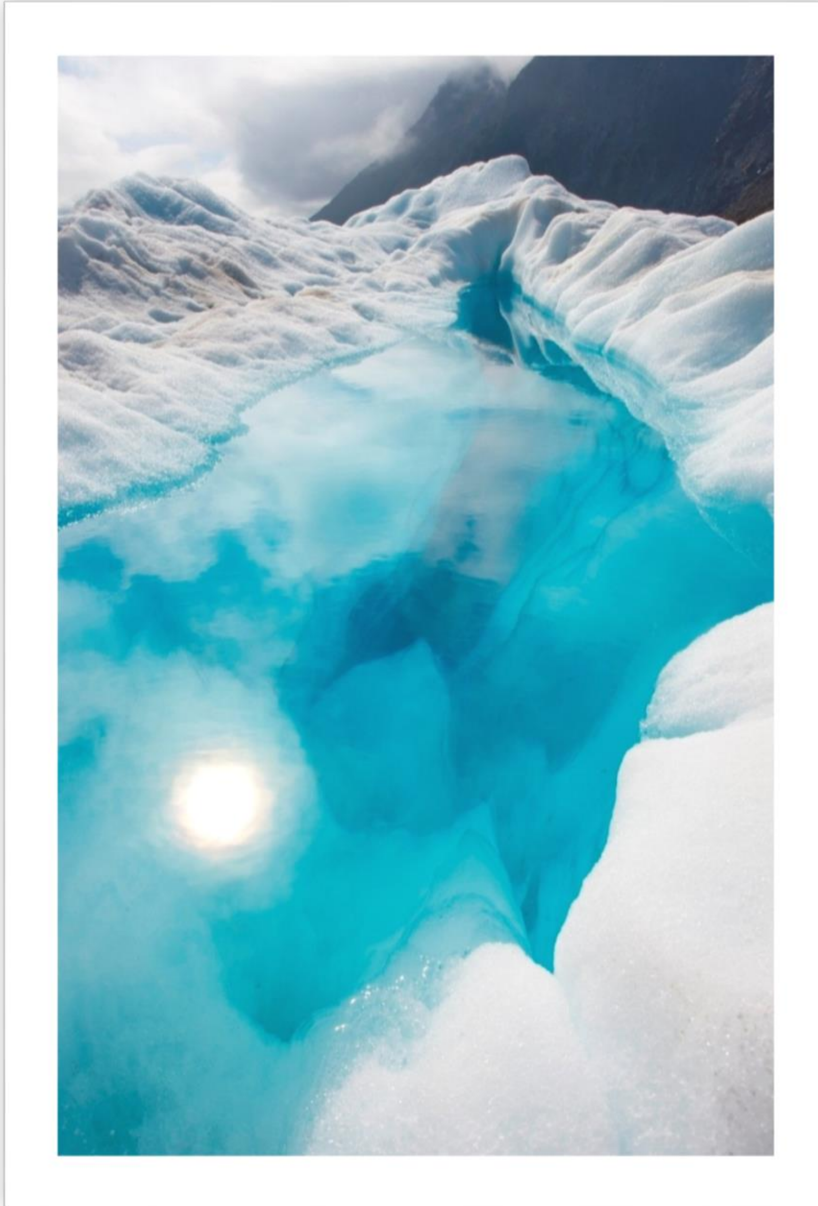


Assignment Report

Hang Wu | ECE516 | 2015 年 2 月 13 日

Table of Contents

SUMMARY OF THINGS DONE

- EEG Data collection for drowsiness in a 3 day period.
- EEG drowsiness classification
- Rewriting real time display code in python
- Combining simulation code between Myo and Muse (unfinished)
- Design the system for the final project of ECE516

EEG DATA COLLECTION AND CLASSIFICATION:

Approximately 68MB of data was collected in a 3 day period from Muse [Figure 1], each takes about 2 minutes (approx. 1500 points). The data was analyzed by Matlab using Neural Network Toolbox [Figure 2]. Please notice that the gradient of 7.44e-5 has reached in only 5 seconds (27 iterations), indicating a fast performance of the algorithm as well as high correlation between input and output. Neural Network provides a 97.9% of overall classification accuracy. [Figure 3] The accuracy for True Positive and True Negative are 97.9% and 98.3%. The regression analysis shows a 98.1% of correlation between actual and simulated output [Figure 4]. This indicates that our method has been really successful for drowsiness classification, and it proves the integrity of the methodology.

Please notice that the algorithm classifies accurately regardless whether the input dataset is complete or not. If the dataset has missing attributes. For example:

0	0.12802	0.11242	0	0	0	0	0	0	0	0
0	0	0	0	0.5845	0.5845	0	1.0108	0.82565	0.2771	0
0	0	0	0	0	0	-0.63876	0.10813	0	0.006648	
1.1483	0.86682	0	0	0	0	0	0	0		

Table 1: Input features of the drowsiness classification algorithm

In this case, the target answer is “alertness.” However, 65% (23 cases out of 43 features) is replaced by 0, simulating the event that poor contact might eliminate many results. The classification result of this set is the following:

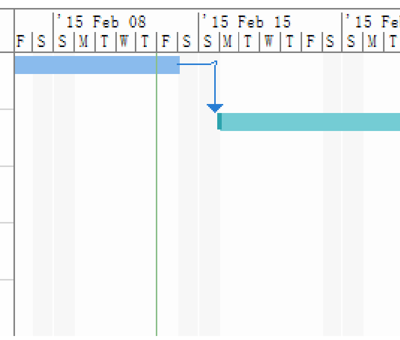
Alert	99.87%
Drowsy	0.13%

The correctness of proposed method is much better than conventional classification/predication method such as moving average, which would produce significantly different result because it's highly influenced by the quality of the data, whereas this method is robust under any circumstances.

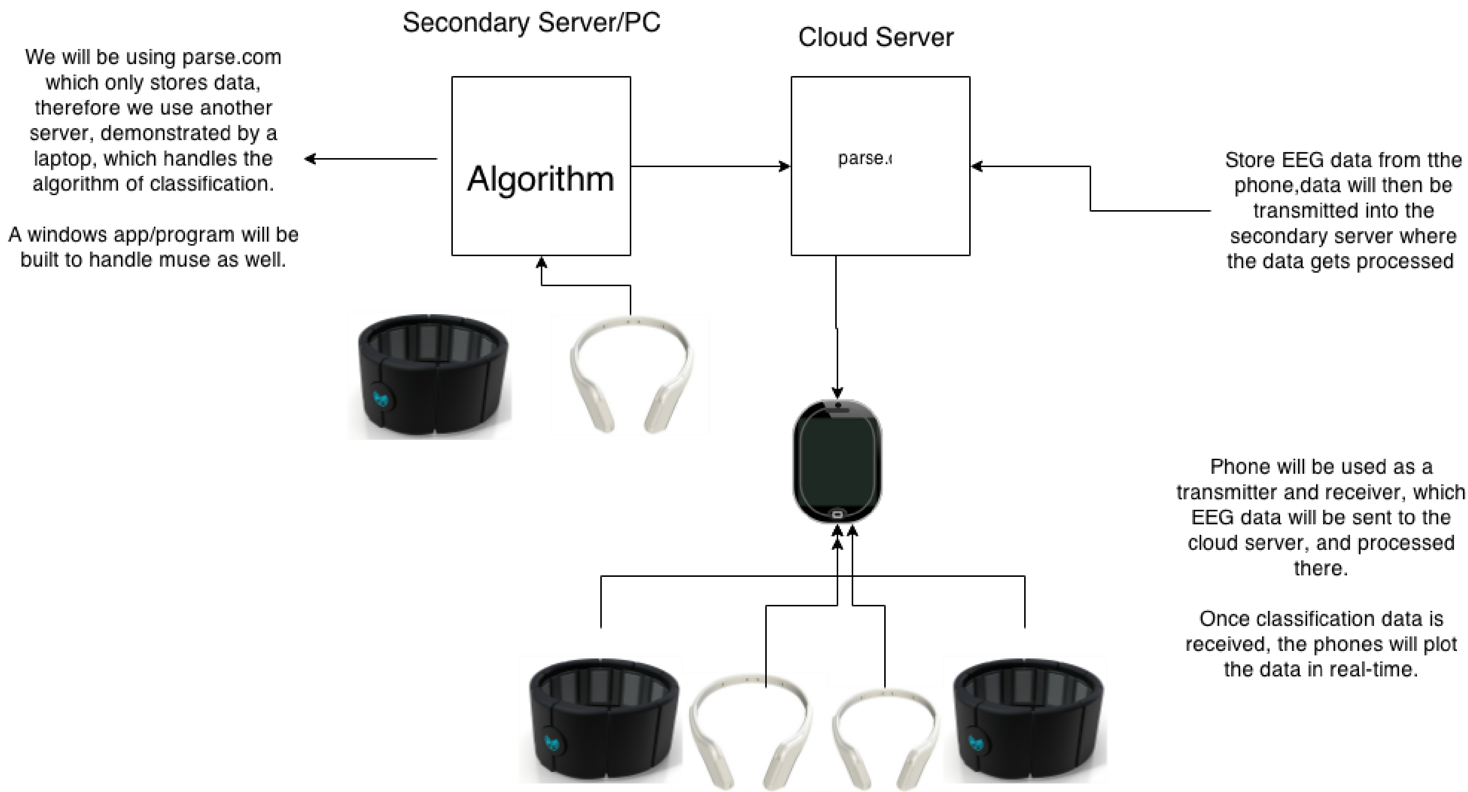
Though one can argue that "overfitting" might have occurred due to the limited data provided, the results suggest that personalized classification (aka, classification using one person's data only) might generate better overall results, with less computational time. This indicates two things: 1). Neural network has a robust performance for classifying emotion, concentration and drowsiness. 2). Classification rate will be better if it's customized individually.

PROJECT TIMELINE

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			myo and muse combination	6 days	'15 Feb 06	'15 Feb 13		
2			Machine Learning algorithm: Muse and Myo	7 days	'15 Feb 16	'15 Feb 24	1	
3			Machien learning Algorithm completed, windows app	7 days	'15 Feb 25	'15 Mar 05	2	
4			Building Mobile App	7 days	'15 Mar 06	'15 Mar 16	3	
5			Systematic Testing and Trials	7 days	'15 Mar 17	'15 Mar 25	4	



SYSTEMATIC DESIGN



awake.mat	2015-02-12 1:10 PM	MATLAB Data	12,574 KB
awake1.mat	2015-02-12 1:19 PM	MATLAB Data	24,822 KB
sleep.mat	2015-02-12 1:14 AM	MATLAB Data	14,907 KB
sleep1.mat	2015-02-12 1:25 AM	MATLAB Data	9,560 KB
sleep2.mat	2015-02-12 1:35 AM	MATLAB Data	9,945 KB
table.xlsx	2015-02-13 12:16 ...	Microsoft Excel ...	8 KB

Figure 1: Matlab data collected from Muse

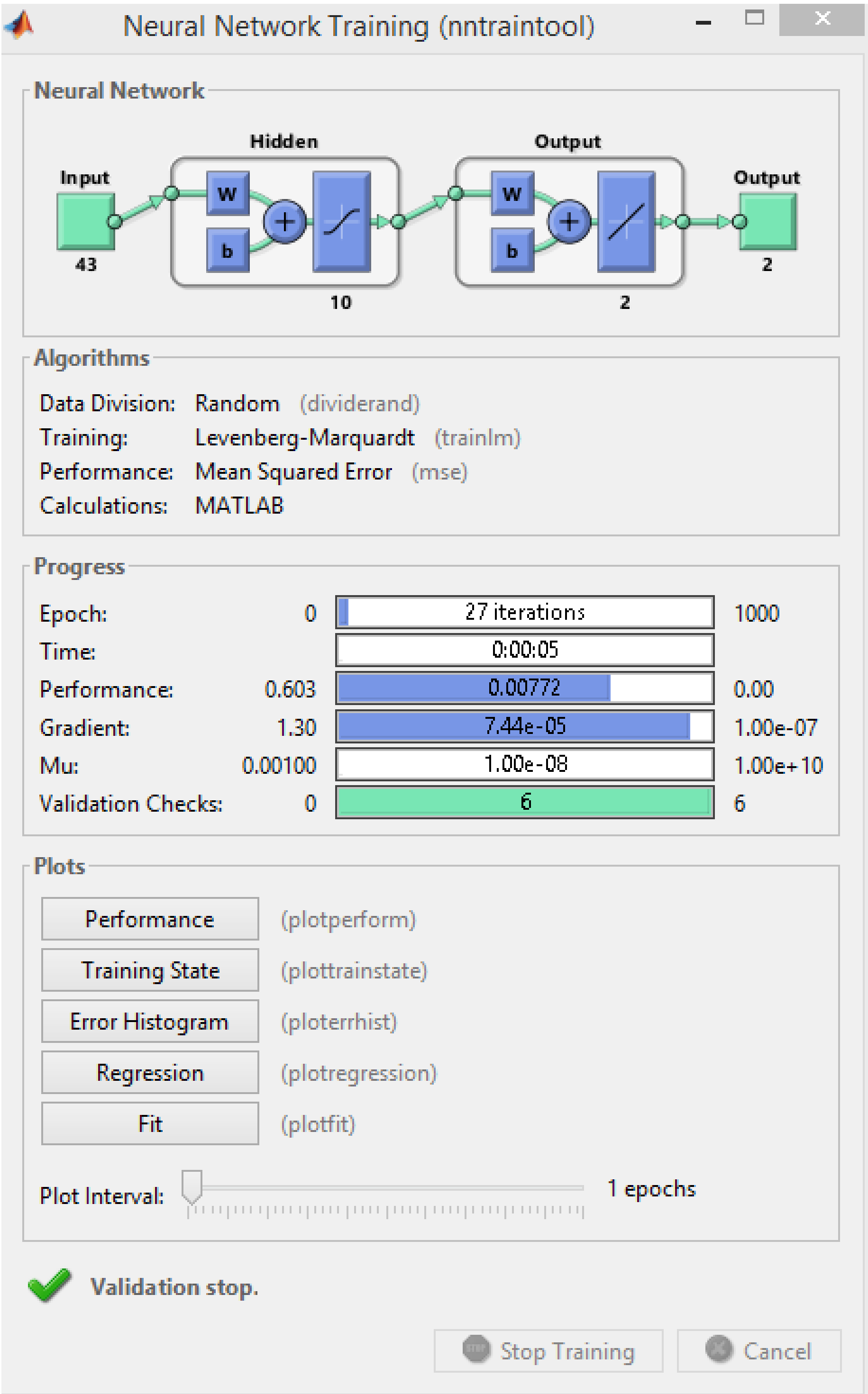


Figure 2: Neural Network Toolbox in Matlab, the training method is Levenbert-Marquardt, and its evaluated through Mean Squared Error.

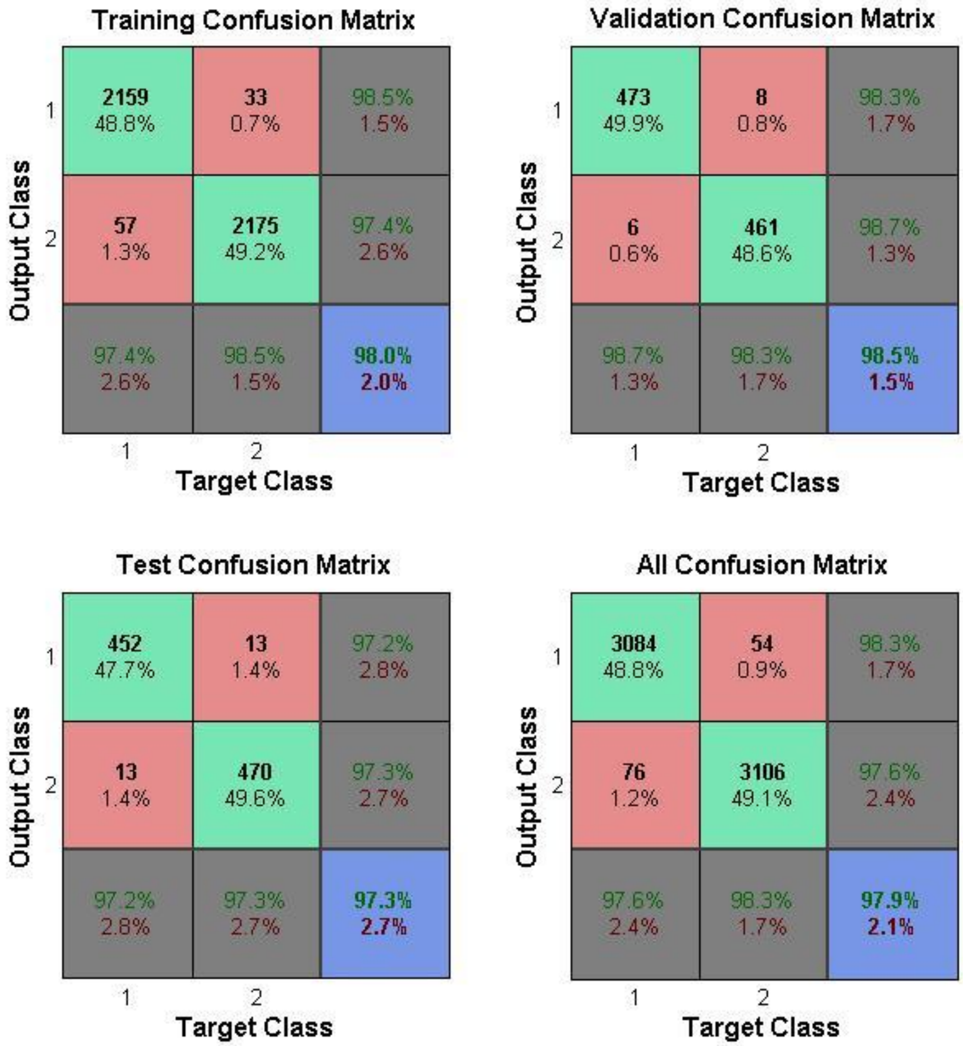


Figure 3: Table of Confusion Matrix, It shows the performance of True Positive, True Negative, False Positive and False Negative. Notice that the overall performance is 97.9%.

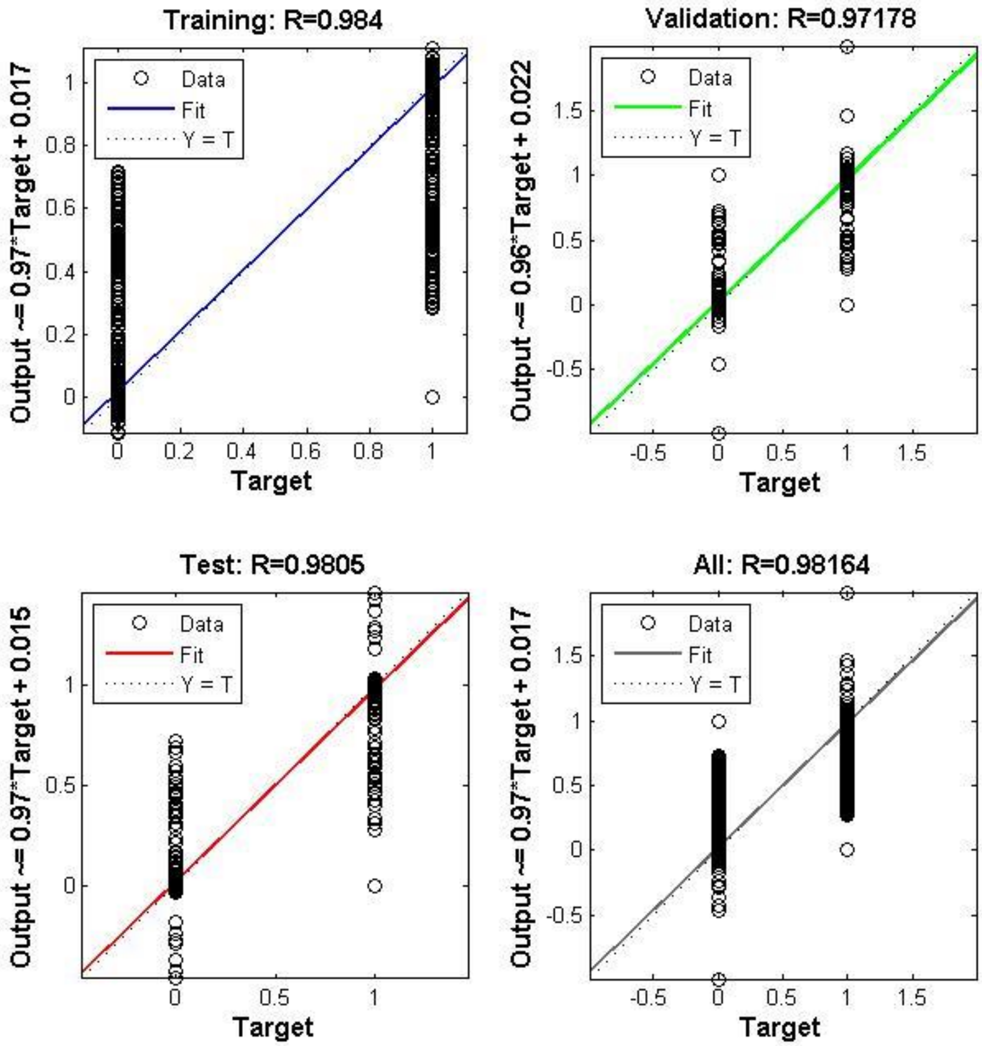
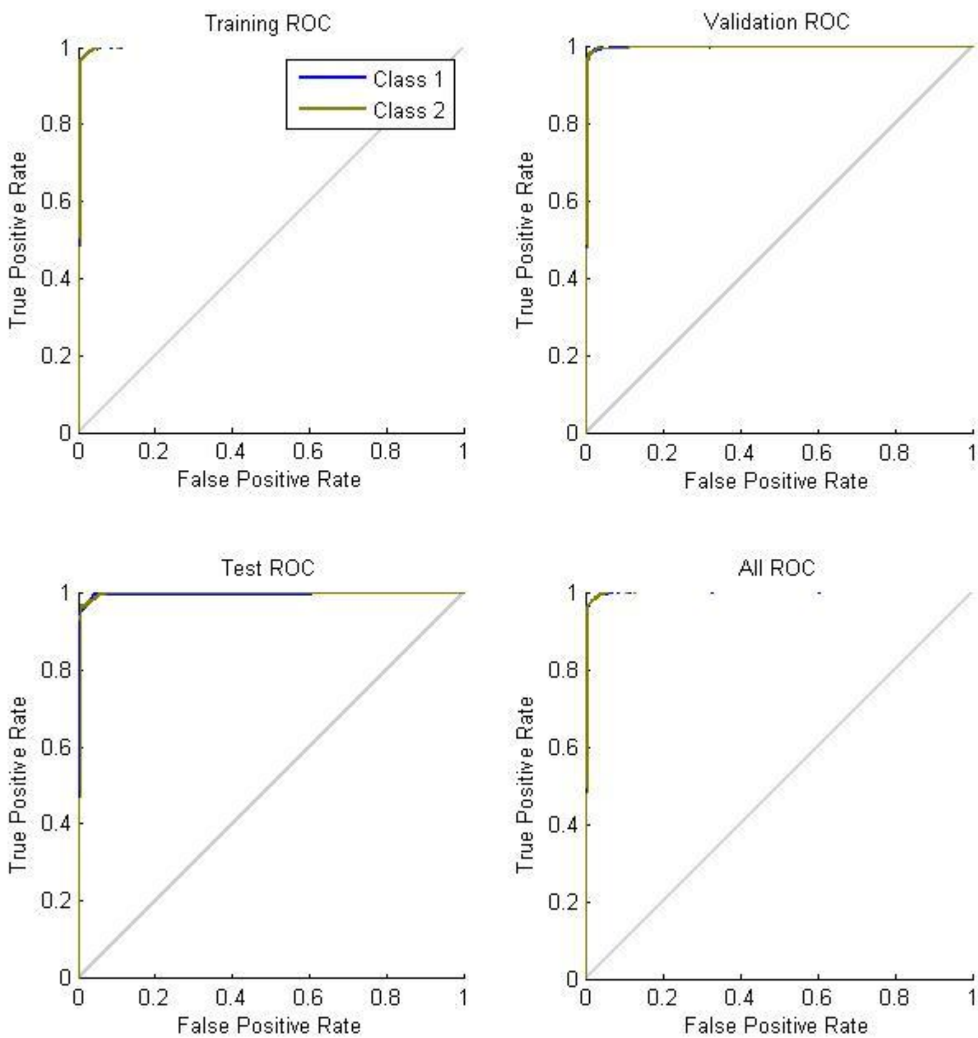


Figure 4: Regression Plot between Target and Output

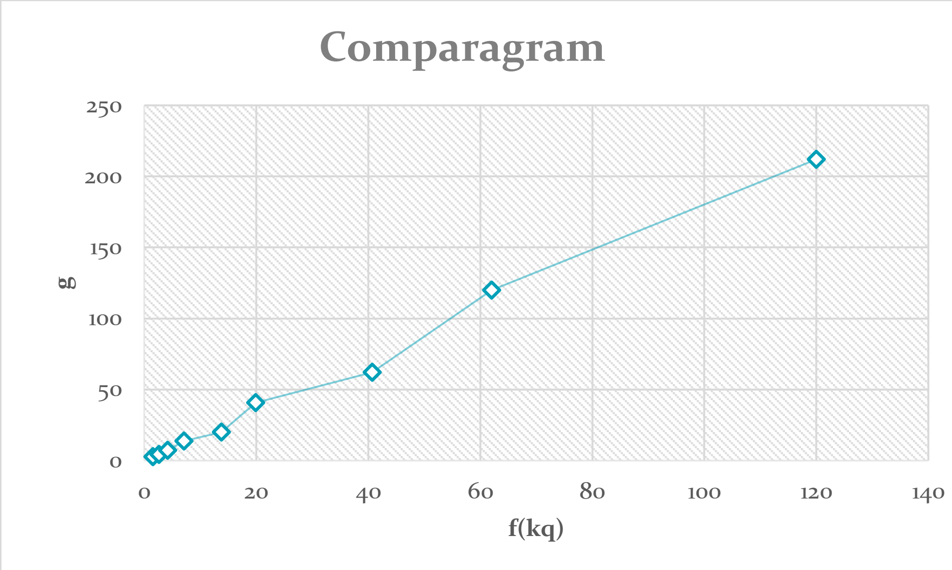


COMPARAMETRIC EQUATIONS

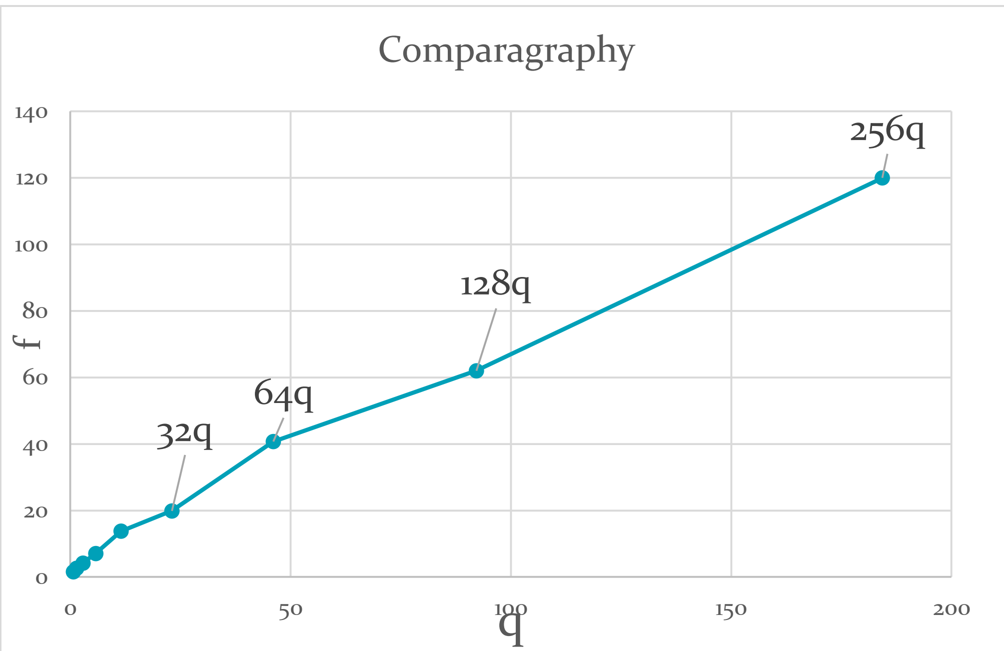
$$g = k^{\gamma} f(1)$$

$$\gamma = log_k \quad \frac{g}{f}(2)$$

By substituting (2), we find
γ *based on experimental data. We use the average to estimate* γ *to be* 0.76. Using the equationf(q) $=$ βq^{γ}, we find the values of q = 0.72. The table of (f, q) is then constructed.



f	g
1.539	2.6
2.6	4.18
4.18	7.07
7.07	13.79
13.79	19.9
19.9	40.7
40.7	62
62	120
120	212



q	f
0.72	1.539
1.44	2.6
2.88	4.18
5.76	7.07
11.52	13.79
23.04	19.9
46.08	40.7
92.16	62
184.32	120

Several insights from the comparagram:

- f(2q) = 1.68 f(q) spotted, which indicates a linear relationship between the quantematrics and their functions.
- The linear relationship can also be spotted on the comparagraphy, where the multiplier of q (2 times) is matched by the multiplier of f (1.74 times).